PUNCTUAL ACTIVITY CLASSIFICATION WITHIN CONSTRAINED SPORTING DOMAINS USING RESERVOIR COMPUTING

DOUG P. L. HUNT

A thesis submitted to
Auckland University of Technology
in fulfillment of the requirements for the degree of
Doctor of Philosophy (PhD)

March, 2017

School of Computing and Mathematical Sciences
Faculty of Design & Creative Technologies

Primary Supervisor: Ass. Prof. Dave Parry
Secondary Supervisor: Prof. Ajit Narayanan
Consultant: Dr. Stefan Schliebs

DECLARATION

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Auckland, March, 2017

Doug P. L. Hunt

ABSTRACT

This research uses Design Science to create a software artefact that employs recurrent neural networks in the form of Reservoir Computing models such as Liquid State Machines and Echo State Networks to classify rare punctual activity in a sports context. This research represents part of a broader plan to use wearable inertial and other sensors to assist in classifying and coaching human movement.

The research is conducted in three main stages with progress reported via four conference papers. The initial stage demonstrates that one form of Reservoir Computing, Liquid State Machines, are capable of classifying "classes" based on synthetic spatio-temporal data. The second stage demonstrates that both Liquid State Machines and Echo State Networks are capable of classifying selected, realistic, punctual human activity normally encountered within equestrian sport. This second stage uses data captured from a wrist mounted inertial sensor used with realistic but scripted activities in laboratory conditions. The third stage utilises data captured from twenty equestrian sports-people undertaking unscripted riding activities in the real world and demonstrates that rare, punctual activity can be successfully classified using an Echo State Network. The real-world data used in the third stage is also captured from a wrist mounted inertial sensor. The punctual activity classified during the third stage of this research represents less than 0.005% of the data captured and so can be said to be "rare".

The main contribution of this research is to demonstrate that it is possible to build a reliable classifier based on spatio-temporal data from punctual human activity using recurrent neural networks in the form of Reservoir Computing models. Reservoir Computing models have been successfully used as classifiers in other areas including speech recognition but have not previously been used to classify human activity. This research concludes that Reservoir Computing models represent a useful adjunct to human activity classifiers but this research does not set out to "prove" that they are necessarily the best or only way of classifying punctual human activities.

A secondary contribution of this research is to extend the differentiation of punctual human activities from durative activities with a cyclic component such as running or rowing and to argue that most prior human activity classification research has focussed on durative activities with much less research focus on short, non-cyclic, punctual activities.

While the classifier artefact developed within this research is intended for use within a sporting context it has other uses beyond this context.

ACKNOWLEDGEMENTS

I have the pleasure to acknowledge some of the many people who have inspired, supported, and educated me over the past five years:

First and foremost, I am very grateful to my primary supervisor *Ass. Prof. Dave Parry* who gave me guidance along the way. I also wish to express my sincere appreciation to my secondary supervisor *Prof. Ajit Narayanan* who has been a source of insight and excellence throughout my research. Many thanks are also due to *Dr. Stefan Schliebs* who wrote the main R libraries for Echo State Networks that I used in my research and who initially provided me with valuable feedback on my research and gave detailed insights into the functioning of the academic world. I am proud to be your friend. Stefan is noted as a "consultant" on the title page. Stefan worked as a contract lecturer within the School of Computing and Maths at AUT University and his role within this research during the initial stages of this work was akin to that of a third supervisor, however, the rules within AUT allow only permanent staff to be PhD supervisors and as Stefan was not a permanent member of staff he could not be acknowledged as a supervisor, instead it was decided that his contribution should be acknowledged via the "consultant" title. Stefan left AUT and ceased active input into this work in 2013, shortly after we produced the paper entitled "*Towards a Wearable Coach: Classifying Sports Activities with Reservoir Computing*".

I am delighted to thank my *wife* for her patient support during my entire education. I will one day, get a real job that pays money! My research has been carried out with the financial support of the School of Computing & Mathematical Science's PhD Scholarship. I thank those members of the school who supported that.

I was born in Mba, Fiji in 1952. Mba is a very small town on the island of Viti Levu. My parents brought our family to New Zealand when I was aged 7 so that my siblings and I could obtain the benefits of a better education system than that which was available to us in Fiji at that time. This thesis and the work done to produce it honours the opportunities that my parents presented to me.

I am proud to be the father of three extraordinary children. My desire to support my children drew my interest firstly into Equestrian sport and then into the idea of building a wearable coach that would help them and others to improve while enjoying their chosen sport.

To Tumbleweed, Chessie, Trinket, Morgan, Polly and even Rusty. The author's horses. Who taught me that with the correct training, a big heart, effort and loyalty anything can be achieved.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

> The human body is continually generating a myriad of bio-signals. These signals, when captured and streamed in real time and in real life, are subject to infinite artifacts. In fact, one of the most important challenges for wearable product developers is to conquer this daunting signal-to-noise ratio for actual, "real life" use cases. Marceau (2014)

The overall goal for this research is the construction of a Activity Classification System (ACS) that successfully classifies the punctual activity Mounting within Equestrian Sport using Reservoir Computing (RC) techniques, based on data from wearable inertial sensors. In particular, the goal is to produce an RC classifier that successfully and reliably classifies the activity of interest based on data captured from riders participating in real, unscripted, riding sessions.

An ACS is a system that is designed to recognise a subset of human activity. In general, the phrase is applied to a computer system of some sort. A common, topical example is the part of the computer code within the Android™ app, Google Fit from Google Inc. that recognises when the person carrying the smart phone running the app is walking, running, cycling or doing a number of other pre-defined human activities.

Human activity recognition using on-body sensors presents a number of challenges (Avci, Bosch, Marin-Perianu, Marin-Perianu, & Havinga, 2010). In many situations this includes a lack of knowledge of overall context such that while it may be possible to distinguish a gesture such as pronating the wrist, it is difficult, without some idea of overall context, to conclude reliably that the gesture is associated with, for example, opening a door by twisting the door handle or turning a car key in a vehicle ignition. One technique used by some researchers to resolve this dilemma is to embed the sensor within some other item such as a car key, pen or baseball bat (Verplaetse, 1996) that has a particular use that constrains the context.

Another technique, using a more generalised, wearable sensor, is to assume or predicate (Lukowicz et al., 2004; J. Ward, Lukowicz, Troster, & Starner, 2006) a particular context or domain so that the choice of meaning of the gesture is constrained by the predicated domain. This is the approach used in this work and in this case the predicated domain is that of Equestrian sport. Sporting domains have some additional benefits as a result of often being strongly defined by rules and traditions.

Another challenge in human activity recognition is the variability in both the spacial and temporal aspects of a particular action both across subjects and even within

a single subject who performs an activity (or action) more than once (J. A. Ward, Lukowicz, & Troster, 2006). Some obvious examples of spatial variability includes differences between left-handed and right-handed activities and differences in technique such as between the drive shot of a professional golfer and an amateur. Examples of obvious temporal differences include taking 75 seconds to mount your horse because the horse is moving away from you or mounting in 7 seconds (or less) as the horse is standing still.

1.1 DEFINITIONS

It is the author's contention that there are different classes of Human activity and that these activities may be broadly distinguished into durative or punctual activities. The author contends further that that at the very least those different classes respond to different approaches when it comes to successfully classifying activities from each class. These contentions are outlined in more detail in the following chapter.

There are potentially a number of different ways of classifying activities into different classes or types but the focus of this research happens to be a temporal one and so it is useful for this research (and perhaps for other researchers) to define the different activity classes in terms of their temporal component. With this in mind the author offers the following definitions of three different activity classes, based on the perceived temporal nature of the activity.

> **Durative** - activities that contain recurrent or cyclic data or data appearing to occur at intervals (e.g walking, running, standing still, rowing, cycling and grooming a horse) and which occur over a longer time.
>
> **Punctual** - short, specific activities that may not contain periodic data (e.g. Pick up a cup, get on a horse, bowl a ball in cricket, hit a ball in baseball and a backhand shot in tennis).
>
> **Complex or Meta** - An activity that is composed of two or more Durative and/or Punctual activities (e.g. Cooking, Riding). In some sense, this definition of complex activity can be seen as almost equivalent to Bobick's (1997) "Action".

These definitions seem to be unique across the activity classification literature based on inertial data and were first published by the author in Hunt, Parry, and Schliebs (2014). However, the concept of temporal based classes for Human activity has been in common usage within the video based activity classification literature and a definition very similar to our own is that of Niebles, Chen, and Fei-Fei (2010, p392–393).

Durative activities usually occur over a longer period of time and have some sort of repeating rhythmic component. Some simple examples of durative activities include running, walking, rowing, grooming a horse and cycling. Punctual activities tend to be shorter and happen once rather than multiple times and so often do not have a repeating rhythmic component to the signal. Some simple examples of punctual activities include bowling a ball in cricket, hitting a ball in baseball, a backhand shot in tennis and mounting a horse. Most of the activity recognition literature looks at

durative activities. This work looks at punctual activities using current state-of-the-art, machine learning, temporal pattern recognition techniques.

1.2 BIG PICTURE — A WEARABLE COACH

This research forms part of a bigger picture, that of creating a wearable coaching system for horse riders. The author believes that activity classifiers are an integral part of the proposed wearable coaching system and so that justifies and positions this work.

The concept of a wearable coach is illustrated in Figure 1.1. The concept includes wearable sensors (**A**), worn by riders as they train. The output from the sensors are fed into a series of classifiers (**B**) that classify the rider's current or emerging activity and in parallel, into an analyser (**C**) that analyses the rider's performance (or style) of the current activity (from B). The Analyser compares the current sensor signals against an ideal model for the current activity and passes this information on to the Decision Support System (**D**) that includes predefined rule-sets (**E**) that establish what feedback may be usefully sent back to the rider. The feedback system (**F**) then provides the feedback to the rider using various on-body techniques (**G**) and the result of the feedback is measured via the sensors (**A**).

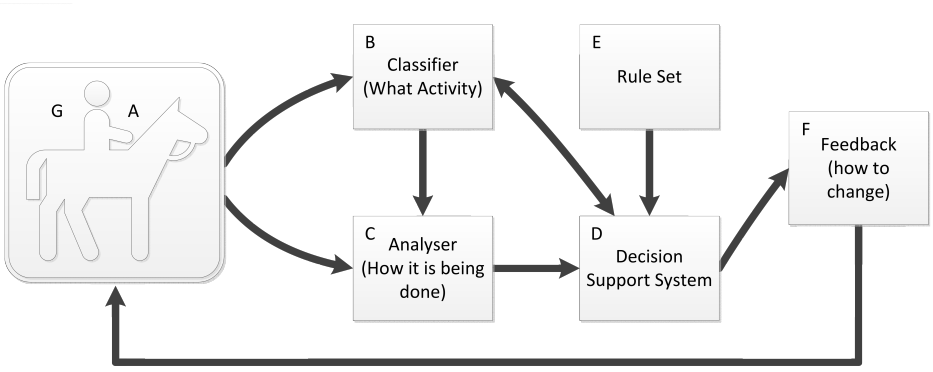


Figure 1.1: Schliebs et al. (2013). Proposed concept of a wearable coach

It is not the author's intention, within this thesis, to argue the pro's and con's of a Wearable Coaching system nor to suggest that the concept of a Wearable Coaching system presented here is the only possible concept or even the best concept. Such arguments, if any, are left for another time and probably another discipline. Within this work it is accepted that a Wearable Coaching system has some (unspecified) merit and given this the conceptual design of such a system described in this document is one possible design that may be workable.

Within this concept, then, item (B) includes a series of classifiers. The job of the classifiers in this situation is to recognise what specific activity is being done by the person using the wearable coach at any specific time. For example, did they just get on their horse (a punctual activity) and are they currently trotting (a durative activity). The author believes that different classifiers will specialise in classifying different

classes of activities and will likely include both Durative and Punctual activity classifiers. Punctual activity classifiers are, it is contended, poorly researched and so the goal of this research is to construct a Punctual classifier that is capable of classifying punctual activities within a riding context, using data from on-body inertial sensors worn by horse riders as they ride. The construction of such a Punctual classifier, it is hoped, will demonstrate that these classifiers are feasible in this situation and so add to knowledge in this area.

1.3 CONTRIBUTION

This work provides two contributions to activity classification based on data from wearable inertial and other sensors. Firstly, it brings across the differentiation between Punctual and Durative activities from video based activity classification and following from this highlights the general lack of sensor-based activity classification research that addresses punctual activities. It also shows that where sensor-based activity classification has been attempted on punctual activities the classification techniques used have primarily been carried across from sensor-based durative activity classification. It shows that where sensor-based durative classification techniques haven't been carried across, the techniques chosen to classify punctual activities have been simplistic, trigger based techniques. This, then, has opened up the question of what other classification techniques are there available that might be suited to classify punctual activities. Classification based on RC machine learning techniques are identified as a possible candidate technique for punctual activities when using sensor-based data.

The second contribution has been to apply RC classification techniques to the successful classification of one particular punctual activity taken from Equestrian sport. RC classification techniques were initially applied to synthetic sensor data as a proof-of-concept, then successfully applied to sensor data collected in a controlled laboratory environment using repeated, scripted activities and lastly, RC techniques were successfully applied to a predefined set of data collected under real world, unscripted conditions. The successful application of RC techniques demonstrates that for the chosen punctual activity and within the pre-defined sensor data set, it was possible to successfully use RC techniques in the way envisaged. Having demonstrated the applicability of RC techniques for a single punctual activity the way is then left open for future research to test if RC techniques are also more generally applicable to other punctual activities and also to test if RC techniques provide some benefit over other classification techniques more often used within sensor-based activity classification.

1.4 RESEARCH OBJECTIVES

The ultimate research objective is to build a working classifier (the artefact) that is capable of differentiating an equestrian stirrup mount event from a base class of everyday, real world horse riding activity. Any instantiation of the classifier beyond the initial proof of concept instance must use the data collected during Hunt (2009). There is usually only a single stirrup mount event of 0.5 to 1.5 seconds duration per

real world riding session of between 40 and 70 minutes duration within the data used in this work and as such the signal associated with the stirrup mount event is relatively rare within the overall real world riding session signal. The rarity can range from 0.004% to 0.125% in the data used within this work. This large discrepancy between the size of the base class and the event of interest means that the area under a Receiver Operating Characteristic (ROC) curve based on the output of the classifier would need to in the order of 0.99 or above in order to adequately differentiate the event of interest.

The artefact will be progressively developed within three phases, described below. During the earlier phases of development less stringent objective measures will be used. These less stringent objective measures are described below, alongside the description of the three development phases. The objectives form the performance measures against which the artefacts developed during the research are measured (as per Design Science).

The research objectives are described in terms of classification and are meant to be inclusive of data pre-processing that will be required to enable more reliable classification, especially when classifying across subjects and places in the real world.

1.4.1 *Phase one research objective*

A proof of concept classifier that is capable of classifying complex, synthetic spatio-temporal data with two synthetic patterns (A and B). To be considered correctly classified, the classifier output is required to be above 1.2 for pattern A and below -1.2 for pattern B for at least 50% of the width of each pattern.

1.4.2 *Phase two research objective*

A classifier that is capable of classifying realistic, non-windowed, scripted punctual activities with reasonably consistent activity completion times. Within this development phase the data captured from the scripted, laboratory based activities from Hunt (2009) will be used. The mounts and dismounts within this data have highly consistent completion times. To be considered correctly classified, the classifier output must be greater than 0.6 on at least one occasion during each event window.

1.4.3 *Phase three research objective*

A classifier that is capable of classifying real world, unscripted stirrup mounts with activity completion times that may not be consistent. Within this development phase the data captured from the unscripted, real world activities from Hunt (2009) will be used. For the stirrup mounts to be considered correctly classified the area under the ROC curve for the classifier output must be 0.99 or greater.

1.5 RESEARCH QUESTION

Is it possible to construct a RC based classifier that is capable of classifying the activities of interest when presented with real, unscripted, un-windowed data captured from a wrist mounted inertial sensor?

This work will use the Design Science research methodology rather than an Experimental research methodology and so the research question will be answered by building a software artefact that meets the research objectives. Consistent with the Design Science methodology, the proposed software artefact will be built using an iterative, incremental process. This process will have three main phases, a proof of concept phase, a phase that classifies the highly regular activities captured from the scripted, laboratory based sessions and a third phase that classifies the stirrup mount activity captured during the unscripted, real world sessions from Hunt (2009). It is the successful completion of this third phase that will ultimately answer the research question. It is worth noting that it is not necessary to show that the resultant RC driven artefact is better than any other technique for classifying this activity but that it is sufficient if it meets the research objectives and it has sufficient novelty that it adds to knowledge in this area. The three phases are described in slightly more detail below.

1st Phase. Construct a RC based classifier that is capable of classifying synthetic, complex, un-windowed spatio-temporal data that meets the research objectives of 1.4.1 as a proof of concept.

2nd Phase. Construct a RC based classifier that is capable of classifying scripted, punctual activities using realistic, un-windowed inertial data with reasonably consistent activity completion times that meets the research objectives of 1.4.2.

3rd Phase. Construct a RC based classifier that is capable of classifying unscripted, punctual activities using real, un-windowed inertial data with possibly inconsistent activity completion times that meets the research objectives of 1.4.3.

1.6 PUBLISHED PAPERS

An iterative approach has been taken to creating the artefacts used within this research. This is consistent with the Design Science methodological framework. As part of this iterative process peer reviewed papers and articles have been published at appropriate times. These papers featured either Liquid State Machine (LSM) or Echo State Network (ESN) approaches to RC and are listed below:

Proof of concept — LSM with synthetic data The conference paper Continuous classification of spatio-temporal data streams using liquid state machines, *In Neural Information Processing (pp. 626–633)* Schliebs and Hunt (2012) reports on tests of a ACS using RC techniques with synthetic spatio-temporal data and concludes that LSM and therefore RC computing techniques are suitable engines for an ACS.

This paper was co-authored with Dr. Stefan Schliebs and as the first published paper within this work served as a learning exercise in academic publishing for Hunt.

Hunt's contribution represents approximately 35% of the effort and includes the (joint) overall plan for the paper; (own) drawing Figure 1; (joint) writing of the results, conclusions and future directions; (own) proof reading; (own)grammatical editing; (own) preparation of the conference slides and the (own) presentation of the paper at the conference. **Schliebs**'s contribution represents approximately 65% of the effort and includes the balance of the paper. This paper has been cited six times including three non-author citations.

A realistic situation — Using LSM with scripted data The paper Towards a Wearable Coach: Classifying Sports Activities with Reservoir Computing, *In Engineering Applications of Neural Networks (pp.233–242)*, Schliebs et al. (2013) demonstrated that RC techniques are able to work with realistic inertial data from scripted activities in a laboratory environment.

This paper was co-authored with Dr. S. Schliebs, Professor N. Kasabov and Associate Professor D. Parry. As the second published paper within this work more responsibility moved to Hunt while brining in small but helpful contributions from two other senior researchers. **Hunt**'s contribution represents approximately 45% of the effort and includes the (joint with Schliebs) overall plan for the paper; (own) section 1, Introduction; (own) section 2, Wearable coach including Figure 1; (own) subsection 3.1, Data including Figures 2 and 3; (joint with Schliebs) section 4, Results; (joint with Schliebs) section 5, Conclusions and Future Directions; (own) proof reading and grammatical corrections; (own) final draft and (own) preparation of the conference slides. **Schliebs**'s contribution represents approximately 50% of the effort including (joint with Hunt) overall plan; (own) Python code for LSM; (own) section 3, Experimental set up and Figure 4; (joint with Hunt) section 4, Results; (joint with Hunt) section 5, Conclusions and Future Directions. **Kasabov**'s contribution represents approximately 4% of the effort including some suggested edits and the presentation of the paper at the conference. **Parry**'s contribution represents approximately 1% of the effort including candidate supervision and organisational support. This paper has one self-author citation.

Using ESN with scripted data & optimised parameters The paper Exploring the applicability of Reservoir methods for Classifying Punctual Sports Activities Using On-body Sensors, *in: Proceedings of ACSC2014, Australasian Computer Science Conference 2014* Hunt et al. (2014), demonstrates a switch from LSM to ESN as the RC based classifier and a switch from grid based search techniques to Particle Swarm Optimiser (PSO) search techniques for the ESN model meta-parameters.

This paper was co-authored with Dr. S. Schliebs and Associate Professor D. Parry. As the third published paper within this work almost all responsibility moved to Hunt. **Hunt**'s contribution represents approximately 98% of the effort and includes the (own) overall plan for the paper; (own) all sections; (own) all figures; (own) proof reading and grammatical editing; (own) preparation of slides and (own) conference presentation. **Schliebs**'s contribution represents approximately 1% of the effort including editorial comment. **Parry**'s contribution represents approximately 1% of the effort including candidate supervision and organisational support. This paper has six citations with five non-author citations and one self-author citation.

Using ESN with real–world data & optimised parameters The paper Using Echo State Networks to Classify Unscripted, Real-World Punctual Activity, *in: Engineering Applications of Neural Networks, Springer, 2015,* Hunt and Parry (2015), represents a switch from laboratory based, scripted activities to real–world based, unscripted riding activities and using a PSO based technique to search for ESN model meta-parameters.

This paper was co-authored with Associate Professor D. Parry. As the fourth published paper within this work almost all responsibility for producing it belongs with Hunt. **Hunt**'s contribution represents approximately 99% of the effort and includes the (own) overall plan for the paper; (own) all sections; (own) all figures; (own) proof reading and grammatical editing; (own) preparation of slides and (own) conference presentation. **Parry**'s contribution represents approximately 1% of the effort including candidate supervision and organisational support. This paper was not cited as of March, 2017.

Explaining the role and importance of classification when developing wearable coaching devices The paper The Role Of Classification In The Development Of Wearable Coaching Devices, *in: Proceedings of MathSport2016, The 13th Australasian Conference on Mathematics and Computers in Sport, Melbourne* Hunt and Parry (2016), presents arguments supporting the importance of classification when developing wearable coaching devices.

This paper was co-authored with Associate Professor D. Parry. As the fifth published paper within this work almost all responsibility for producing it belongs with Hunt. **Hunt**'s contribution represents approximately 99% of the effort and includes the (own) overall plan for the paper; (own) all sections; (own) all figures; (own) proof reading and grammatical editing; (own) preparation of slides and (own) conference presentation. **Parry**'s contribution represents approximately 1% of the effort including candidate supervision and organisational support. This paper was not cited as of March, 2017.

1.7 THESIS STRUCTURE

The structure of the thesis follows the research objectives and is outlined below.

Chapter 2 – Problem Background

This chapter describes the problem background. It starts by briefly reviewing the wider area of wearable coaching devices to provide context for what follows. It also reviews the domain of Equestrian sport and the concept of domain constrained activity classification.and reviews current research in connected areas that are related to the problem space.

Chapter 3 – Literature Review

This chapter reviews current developments in the area of activity classification, especially activity classification based on wearable inertial (or other) sensors. In particular, this work differentiates between punctual and durative (cyclic) activities and between punctual activities with regular temporal frames and those punctual activities with irregular temporal frames.

Next current research in durative activity classification based on wearable sensors is reviewed. This is then followed by a review of current research in punctual activity classification based on wearable sensors. There is also a brief review of wearable versus off-body sensors for activity classification.

Chapter 4 – Methodology & Data

This chapter provides a description of the approach to Design Science used within this research and the iterative development approach that was followed as part of that methodology. This chapter also describes how the data used in this work was validated and pre-processed. Including issues and characteristics of the data.

Chapters 5 and 6 – Design Cycles

The three phases of the design cycle are covered in these two chapters. This includes the early proof of concept work using LSM and synthetic data; the move from a LSM to an ESN model; the second phase utilising an ESN and the laboratory situated activity data and finally the real world situated activity data using an ESN.

Chapter 7 – Discussion

A discussion of issues that were encountered during this research.

Chapter 8 – Conclusions and Future Work

The conclusions of the research and some ideas on future work that can extend the ideas introduced within this research.

1.8 PRESENTATION AND THE AUTHOR'S VOICE

The diagrams presented within this document are generally sized so that they flow within the text in a natural way. This means that sometimes the details within the diagrams are hard to pick out for a reader not completely familiar with this research. To assist these readers, the diagrams are also available as stand alone (electronic) documents in a larger form. In addition, if this document is being read in PDF format then it is possible with almost all figures to use the PDF reader's capabilities to zoom in on details with the diagrams. Some diagrams, however, are presented in a manner that highlights their overall characteristics and so information may be obscured if the reader tries to zoom in on all diagrams.

Throughout this work a number of key people including my supervisors have assisted and have made contributions, especially at the beginning of the work. These people have been acknowledged within the Acknowledgements Section, however, my own attitude towards teams and teamwork is reflected via my "voice". In the beginning of these writings and during the initial design cycle phase I sometimes use the words *we* and *our* rather than *I* and *my*. I do this to personally acknowledge the value that I place on the contributions made by others and to affirm my commitment to teamwork. The use of *we* and *our* is uncommon within academic writing and so I hope that the reader is prepared to bear with me in my use of these words. In the discussion and conclusion chapters when I am offering my own opinion I revert to using the third person in my writing. Please understand and forgive the change. Regardless of my language, all errors, omissions and inconsistencies are my own responsibility.

1.9 SUMMARY

This research is designed to construct a Recurrent Neural Network (RNN) model that successfully classifies punctual human activity based on real–world activity data collected from equestrian sports people. It is constrained to working reliably within the domain of equestrian sport. It uses a design science methodology and sets out to design, build, test, tune a working system using an iterative approach. As a consequence of the structure of the RNN model that has inherent temporal "memory" the data used is unwindowed inertial data captured via a commercial Inertial Measurement Unit (IMU).

This research's original contribution is to successfully apply RNN classification techniques to a selected punctual human activity using simple features extracted from spatio-temporal inertial data and to carry the distinction between Punctual and Durative activities across to inertial data based activity classification from video data based activity classification.

Chapter 2

PROBLEM BACKGROUND

This chapter describes the problem background and highlights some of the literature in connected areas that are related to the problem space.

2.1 INTRODUCTION

This chapter starts by briefly describing the Punctual activities of interest within this work. This is followed by the description of the concept of a wearable coach and some of the areas of research associated with this concept including Adaptive Systems, context as an aid to classification and in particular constrained contexts, how successful classification can assist with inferring future activities and some additional justifications for an ACS as a component of a possible wearable coaching system based on on-body inertial sensors. This then leads to a quick look at on-body versus off-body activity sensing and the use of on-body inertial sensors for purposes other than classification. This is then followed by a brief summary and introduces the next chapter that reviews some current literature relating to inertial based activity classification systems.

2.2 ACTIVITY DEFINITIONS

For this work, the activity definitions are:

> **Mount** – A stirrup mount from the time when a rider with one leg in the stirrup, lifts the second leg off the ground or mounting block in order to mount until the time when they are seated in the saddle.
>
> **Dismount** – The time from when a rider leans forward (prior to dismounting) until they are standing on the ground.

As with all activities, identifying the precise beginning and ending of a mount/dismount is not trivial (Plötz, 2010) and so a definition of the activity with a recognisable start and end is required.

These definition were developed by working backwards from the point where most observers would agree that a rider is mounted (or dismounted) on a horse. For a Mount this endpoint is the point when the rider first puts their weight into the saddle or both stirrups. Working backwards from this, a point was found where there was

reasonable consistency across riders. This point was the time when the rider lifts their right leg off the ground for the last time, prior to mounting.

Mounting a horse via a stirrup mount can be a strenuous endeavour and for a shorter rider or a taller horse, the rider may need to bounce a number of times before they accumulate enough impulsion to lift their own weight and body up and over the horse. By starting the definition after any bounces, it is possible to encapsulate that part of the mounting sequence that is most consistent while also encompassing the essence of the mount.

Similarly, with the Dismount the endpoint is when the rider is standing on the ground after dismounting and working backwards from this point to when they lean forward (in order to rebalance their weight in preparation for dismounting).

Neither of the definitions is inclusive of all possible classes of either mounting or dismounting but the author considers them sufficient to cover a reasonable percentage of the mounts and dismounts that are likely to be encountered in equestrian sport. As far as the author is aware, there are no other generally accepted definitions for these activities, for the purposes of activity classification.

A more detailed description of these activities is included below to assist any reader that does not have a detailed knowledge of the horse riding domain.

2.2.1 *Mounting*

Mounting a horse using a stirrup mount technique often follows the following process:

- Place (left) foot in stirrup (by definition, order may vary)
- Place (left) hand on saddle pommel (placement may vary, order may vary)
- Place (right) hand on saddle cantle (placement may vary, order may vary)
- Lift (right) leg off the ground (speed may vary, initial “hops” may be involved)
- Swing (right) leg over the cantle (in parallel with moving the right hand)
- Move (right) hand off cantle (in parallel with the right leg going across the cantle)
- Sit into (put weight into) the saddle (may be delayed for some time)
- Place (right) foot into stirrup (optional)

Figure 2.1 shows a series of photographs of a rider following this typical sequence. The definition of Mounting would have this rider starting their stirrup mount somewhere between images three and four (counting left to right, top to bottom) and ending with image six.

Note that some aspects of Mounting and Dismounting are rigorously standardised throughout European riding, internationally. Within European riding, the rider **always** mounts and dismounts from the horse’s left side (as depicted in the above images). The horse’s left side is also colloquially called the ”near side”, for this reason. To mount or dismount from the opposite or ”off side” within European riding is considered a ”fault” and would disqualify a rider competing in any of the European

Figure 2.1: Example of a participant mounting a horse

riding disciplines. As a result, it is so rare that most horses trained for European riding disciplines would be confused by a rider trying to mount or dismount from the off side. Hunt (2009) provides background on why this tradition is so rigorously encountered within European riding.

2.2.2 *Dismounting*

Dismounting a horse often follows one of the two following processes. For a stirrup dismount:

- Remove (right) foot from stirrup (order may vary).
- Lean forward (order may vary).
- Place (left) hand on saddle pommel or the horse's neck (placement may vary, order may vary).
- Take the rider's weight onto the (left) leg and stirrup, then lift the rider's body slightly.
- Lift (right) leg up and swing it over the saddle cantle (speed may vary).
- Move (right) hand and place it on the saddle cantle (optional, speed may vary).
- Lower the rider's body while placing the (right) leg under the rider's body until the (right) leg touches the ground (speed may vary).
- Take the rider's weight onto the (right) leg.
- Remove (left) foot from stirrup and stand upright.
- Remove hands from the horse and saddle (speed may vary).

For a drop dismount:

- Remove both feet from stirrups (order may vary).
- Lean forward further than for a stirrup dismount (order may vary).

- Place (left) hand on saddle pommel or the horse's neck (placement may vary, order may vary).
- Take the rider's weight onto the hip.
- Lift (right) leg up and swing it over the saddle cantle (speed may vary)
- Once the rider's (right) leg is across the cantle, bring both legs together under the rider's body.
- Drop the rider's body until both legs touch the ground, while '*sliding*' off the saddle flap (speed may vary).
- Take the rider's weight onto both legs (speed may vary).
- Bend at the knees to cushion the shock and then stand upright (speed may vary).
- Remove hands from the horse and saddle (speed may vary).

The author believes that mounts and dismounts can be categorised as Punctual rather than Durative activities (Please see section 1.1 where Durative and Punctual activities are distinguished). It is also possible that they could be categorised as meta-activities, comprising a number of shorter Punctual activities and the definitions of these activities supports this possibility (E.g. Mount - left foot in stirrup; left hand on saddle pommel; right hand on saddle cantle; lift right leg off the ground and swing it over the saddle; move right hand off saddle cantle as leg crosses the saddle; sit into the saddle). However, in normal situations, these movements happen so quickly and in such an integrated manner that most observers would consider them to be a single, Punctual activity and in addition, at this stage of the work, there is no wish to address meta-activities and so this choice is out of scope.

Based on the author's earlier Masters work it is believed that for stirrup mounts, at least, there is a short and common enough signature that a Punctual ACS will successfully classify this activity. In addition, the author's domain expertise leads him to expect that dismounts are also a good candidate for classification via a Punctual ACS because dismounts tend to be quick, consistent and relatively simple at their core.

2.3 BIG PICTURE – A WEARABLE COACH

Section 1.2 and Figure 1.1 describe the concept of a Wearable Coach. A Punctual activity classifier is one of several classifiers that are necessary within this system and which, together, comprise one of the major components that make up the wearable coaching system.

A brief review of the wider area of wearable coaching systems follows, to provide context for the wider picture. The wider picture of the wearable coaching system is important because, provided it is feasible, it suggests that any necessary components, such as an activity classifier are important for delivering the wearable coaching system. The author's own concept of a wearable coach has several ACS as an integral part of the overall system and so gives justification for doing this work.

A number of authors have suggested wearable coaching systems of various forms. A very early pioneering researcher, Verplaetse (1996) had the vision to suggested that everyday objects would be developed that use inertial sensors to provide self-knowledge of motion and that these objects would then use that knowledge to communicate useful information to the people/person using the object. Examples given include a baseball bat with a built in inertial sensor that could tell a batter how well he was hitting the ball. Although this is not a wearable system, it nevertheless encompasses very similar ideas and helped set the scene for following researchers.

One of the first mentions of an automated coach comes from Bacic (2004). While this author did not at that stage suggest that this would be a wearable coach, nevertheless the proposed concept of an automated coach for tennis based on segmented video data that is analysed for regions of interest using explicit rules encompasses many of the same concepts.

Kawahara et al. (2005) comes closer to the concept of a wearable coach when they proposed a Wearable Exercise Support system and called their activity classifier a "*Context Inference*" system. While this "E-Coaching" system is designed to provide feedback, the feedback is envisaged as suggestions to rehydrate and suggestions for further exercises rather than feedback designed to improve the athlete's technique.

Several groups have reported on wearable systems designed to support a single, pre-defined activity. Buttussi and Chittaro (2008) described a wearable system (MOPET) for assessing the user's ability to perform a single, pre-defined, model exercise activity. As such, it neither requires a classifier (as it only applies to a single activity) nor does it propose to provide feedback. Ghasemzadeh, Loseu, and Jafari (2009) proposed a wearable coaching system for golf driver swings, but within this system the authors manually classified activities and so this system, as described, is incapable of providing real-time feedback on techniques. Similarly, Ahmadi, Rowlands, and James (2009) proposed a wearable coach for the first serve in tennis and in this case no classifier was required as only a single activity was considered and pre-selected. Wixted, Portus, Spratford, and James (2011) described a wearable system for detecting a throwing action in cricket where the authors manually classified and segmented activities of interest.

Connaghan et al. (2011) proposed a wearable tennis coach that classifies tennis strokes so that they could be counted by a coach and perhaps assessed off-line. This concept is essentially an automated auditor and is similar to commercial wearable systems such as Fitbit® which count steps and energy expenditure.

Spelmezan, Schanowski, and Borchers (2009) reported on a wearable coaching system for snowboarders and skiers, the authors described successful attempts to classify turns as being front-side edge or back-side edge turns using force sensors in the athlete's boots. At the stage of reporting this work, the proposed system was capable of classification but was not capable of analysing or providing feedback on technique. In a similar vein, Bachlin, Forster, and Troster (2009); Bachlin and Troster (2011) reported on a wearable swimming coach called "*SwimMaster*", that automatically detected (classified) wall-push-off, wall-turn and wall-strike events but was not at that stage capable of analysis or feedback.

Baca (2012) described a more general Mobile Coach that is a combination of wearable sensors and a back-end server system where the wearable sensors collect information that is communicated back to the remote server system. The authors do not address real-time feedback to the athlete within this concept.

Hunt (2009) first, publicly, proposed a wearable riding coach. This concept was described in more detail in Schliebs et al. (2013) and further refined in Hunt et al. (2014).

From this collection of research it is concluded that a number of other researchers have recognized that a Wearable Coaching system is feasible in one form or another and has utility. This is taken as evidence to justify that the concept of a wearable coach is feasible and possible within current technology. This, in turn, justifies this work on activity classifiers.

2.3.1 *The Conceptual Design of a Possible Wearable Coach*

The concept for a Wearable Riding Coaching system is illustrated in Figure 1.1 on page 3 within Section 1.2. In the concept of a wearable riding coach, the (human) athlete wears sensors on various parts of their body (**A**) as they train. Those sensors feed output in parallel to a series of Activity Classifiers (**B**) and Analysers (**C**). The purpose of the classifiers is to classify the rider's current activity and, perhaps, in conjunction with the Decision Support system (**D**) to make a best guess at the next or emerging activity. The author envisages multiple classifiers that may be focussed on classifying different classes of activities.

The classifiers then pass the identification of the current activity and, perhaps, the emerging activity onto an appropriate analyser (**C**) that can then analyse either the rider's performance (or style) on the current activity or on the emerging activity. The Analyser compares the current sensor signals against an ideal model for the current activity and passes this information on to the Decision Support System (**D**).

The Decision Support System would then use the outputs from the classifier(s), Analyser(s) and some sort of Rule Set (**E**) to decide if any coaching feedback is required and if so to prioritise the feedback so as to best achieve the coaching goals for the current training session. A feedback scenario (**F**) would be selected and the desired coaching feedback would be delivered to the athlete via on-body feedback techniques (**G**) and the result of the feedback is measured via the sensors (**A**) the analyser (**C**) and decision support system (**D**).

Based on this design concept, a series of ACS's (component **B** in figure 1.1) are required in order to construct the proposed wearable coaching system for horse riders. For example, it is envisaged that ongoing activities with a rhythmic component to the signal may be classified using a classifier(s) designed for Durative activities whereas shorter activities without any particular rhythmic or repetitive component may be classified based on classifier(s) that are specifically designed for Punctual activities.

In addition, it is thought that some complex activities that are composed of a series of smaller, sub-activities where the sub-activities could be Durative, Punctual or mixed could be considered as meta-activities and perhaps classified using a Hidden Markov Model (HMM) or similar technique designed for stateful classification. Such

meta-activities may be easier to classify by first classifying the sub-activities and then stringing those sub-activities together in some way to form a syntax of sorts. This would be especially useful when the sub-activities may be performed in differing orders, where the sub-activities may vary or when some sub-activities are optional.

The author believes that mounts and dismounts are best categorised as Punctual rather than Durative activities and that is the activity category that is used within this work and so the classifier that will be developed will be designed solely for Punctual activities. In addition, an attempt will be made to classify real world Punctual activities without using additional, non-inertial sensor or other data.

2.3.2 *Adaptive Systems and Ambient Intelligence*

The proposed wearable coaching system is an adaptive one and so it is useful to look a bit wider in the literature at other adaptive electronic systems to gain some perspective. One such area that has closely related adaptive systems is the area of Ambient Intelligence (AmI).

Aarts and Wichert (2009, pp. 1) define AmI as "... *electronic environments that are sensitive and responsive to the presence of people*". AmI is a recent area of research for Human-Computer interface issues and is part of Ubiquitous Computing and Context Recognition. In this context, being sensitive to the presence of people goes well beyond merely sensing that a person or even a particular person is present but possibly also includes knowledge of what that person is doing. That is, a knowledge of their activities at each moment. An ACS would then seem to be an important element of any system that purports to adapt to what a human being is doing at a particular time, thus reinforcing the importance of developing comprehensive ACS systems. The history in this area goes even further back with some seminal work from Weiser (1993) setting the scene with his ideas for adaptive computer systems. Pantic, Pentland, Nijholt, and Huang (2007) extended Weiser's work and they highlight the importance of activity classification in their work.

2.3.3 *Context as an aid to classification*

A knowledge of context can also add a significant dimension, for example, it may be generally insufficient for many purposes to simply know that a person is running (activity). But a knowledge of the wider context can add additional, important information that drives the system to provide relevant advice. While not directly concerned with activity classification, Abowd et al. (1999) reiterates the importance of contextual information in effectively situating computer systems so that they are more usable. Abowd et al. (1999) was particularly interested in wearable and ubiquitous computing and so was concerned about designing these systems so as to enable them to have some understanding of and be adaptive to human context.

More specifically, if a person is running to catch a bus (in this case knowledge of where the person is, where the bus is, where the bus stop is, the person's current speed, the buses current speed and both their probably maximum speeds may enable the system to provide advice on if the runner will reach the bus stop in time to catch

the bus or not); or running for exercise (in this case knowledge of the distance the person has already run, their probable energy usage since the start of the run, their probable route for the rest of the run and their exercise goals may be used to give advice on run tempo, cadence and remaining distance to run in order to achieve the exercise goals).

A system may be able to infer context from a synthesis of the current activity plus additional information. Considering the earlier example of the person who was running, a system may be able to recognise that this is the approximate time that the person usually goes to work; that today is a work day; and that the person is on their normal route between their home and the usual bus stop that they use. From this the system may infer the context that the person is running to the bus stop in order to catch the next bus.

A helpful system could then (with a knowledge of bus routes and timetables) calculate that the next due bus on that route has not yet reached the person's usual bus stop and pro actively encourage that person to continue running at their current pace (or suggest a speed up or a slow down to a walk if there is no chance of getting to the bus stop in time). Such a system would probably be very useful (at least for me). However, systems similar to this such as Google's Google Now often struggle to infer the correct context in many except the most obvious situations.

Alternatively, the additional information could be supplied to the system in the form of a static, domain constraint. Taking the second running situation as an example, where the person is running for exercise, it would be simpler for the system to infer that the person was running for exercise, even though they were currently on the same route between their home and the bus stop if prior to the activity being detected, the person had placed the system (such as their smart phone) into 'exercise' mode. Putting the system into 'exercise' mode places a domain constraint on the system and may assist with both the correct classification of the activity and the inference of context.

Bachlin and Troster (2011) "*SwimMaster*" system is such a domain constrained system that could be used with a swimmer training for the medley to trigger a change of model for swimming style when a "*wall-turn*" event is detected. This useful system would not be so useful outside of the competitive pool swimming domain. For example, an ocean swimmer who trains in the ocean is unlikely to encounter (or even need to know about) wall-turn events.

Unconstrained activity recognition presents a number of challenges including a general lack of overall context in many situations that makes it difficult to distinguish two similar movements (*e.g.* turning a door knob to open a door and turning a key to start a vehicle). Most researchers have recognised that developing an unconstrained, generalised activity classifier that is capable of classifying any or almost every human activity is unachievable at this stage and so almost all classifiers that are reported on are either explicitly domain constrained or are implicitly domain constrained. Domain constrained activity recognition, however, has been shown to be more achievable and reliable Lukowicz et al. (2004); Stiefmeier, Ogris, Junker, Lukowicz, and Troster (2006). Constraining the domain to a particular sport (such as equestrian sport) is also potentially useful, particularly if the rules or traditions of that sport add

further activity and style constraints. Some examples of Domain Constrained activity classification are shown in Table 2.1 and includes a complex example, Veltink, Bussmann, de Vries, Martens, and Van Lummel (1996), where the explicit domain is the seven activities listed and the implicit domain is constrained by the specific equipment used (stairs and bicycle) and the scripted/situational constraints such as their participants walking without interruption along a particular corridor; and an attempt to develop a generalised system, Van Laerhoven and Cakmakci (2000), which is ultimately constrained to those activities that can be distinguished by their system.

Research	Domain Constraint	Type
Veltink et al. (1996)	Seven activities	Explicit domain
Randell and Muller (2000)	Seven activities	Explicit domain
Van Laerhoven and Cakmakci (2000)	Activities distinguished by the system	Implicit domain
Bacic (2004)	Tennis	Explicit domain
Kawahara et al. (2005)	Exercise	Explicit domain
Buttussi and Chittaro (2008)	A single predefined activity	Implicit domain
Bachlin et al. (2009)	Pool swimming	Implicit domain
Ghasemzadeh et al. (2009)	Golf driver swings	Explicit domain
Spelmezan et al. (2009)	Skiing and snowboarding	Explicit domain

Table 2.1: Classification of Activity Classification Systems by Domain Constraint

At this stage of development, the author's research interests lie within this area of domain-constrained activity classification.

2.3.4 *Inferring next activity*

Conceptually, a domain constrained system with sufficient knowledge of its domain and the ability to classify current activity might be able to infer the next probable activity and provide coaching advice specifically for that activity, as it is being performed. Thus correct activity classification may give insights into intention and intended activity as well as simply identifying the current activity.

For example, a sport coaching system that has been placed into 'right-handed tennis' mode that then detects that a tennis player has raised their right arm over their left shoulder may classify the current activity as 'preparing for a back hand shot' and so as the players right arm starts to move forward it may infer that the person is performing a 'back-hand shot' and be then able to assess and perhaps provide feedback on the quality of that shot as the player is making the shot.

More general activity recognition then, such as the work on classifying Mounts within horse riding, may provide contextual information that allows more detailed activity recognition and may have predictive value with regard to future events.

Successfully classifying a mount activity enables a system to infer that the rider is on the horse and possibly ready for coaching advice. While successfully classifying a dismount activity provides strong inference that the rider is no longer interested in riding coaching advice (as they are no longer riding).

2.3.5 *Additional justifications for an ACS*

Human activity recognition is an important prerequisite in many other areas of endeavour such as auditing, quality control and non-sports coaching within areas such as medicine.

- auditing – Did the nurse unscrew the cap of the pill bottle? Did the patient do their physiotherapy exercises when at home?
- manufacturing quality control – Did the car assembly worker fit the part using the specified technique?
- medical coaching – Did the trainee surgeon use the scalpel with the recommended pressure?

In all these examples and in many other situations knowledge of what activity a person is currently doing (or has done) is important. For automated or computerised systems an ability to automatically recognise human activities (as opposed to a human having to manually tell a computerised system what the activity is/was) provides useful information and context and may be an essential element of usability or even feasibility for the system as a whole.

2.3.6 *Big Picture Summary*

The conceptual model of a wearable riding coach requires a set of classifiers and so provides justification for developing them but even outside of that particular conceptual model, there seems to be recognition from other researchers for the need for activity classifiers in general as demonstrated by other work in this area such as Baca (2012); Bachlin and Troster (2011); Chuang, Wang, Yang, and Kao (2012); Connaghan et al. (2011); Harms, Amft, TrÃűster, and Roggen (2008); Kawahara et al. (2005); Mitchell, Monaghan, and O'Connor (2013); Novatchkov and Baca (2012); Qaisar et al. (2013); Spelmezan et al. (2009); Taylor, Abdulla, Helmer, Lee, and Blanchonette (2011).

In fact (Baca, 2012, p. 2) states explicitly that "*One main basis of almost any intelligent feedback system or adaptive system is the successful recognition or classification of patterns underlying the human motion just performed*".

Bobick (1997) defined three levels of motion understanding with increasing levels of required knowledge and labelled them *movement, activity and action*. Thus placing activity classification at the centre of understanding human action.

This is taken as evidence to support the contention that relevant classifiers are an important component of the proposed wearable coaching system and this justifies research in this area.

2.4 DATA SENSING LOCATIONS

Existing activity, movement and gesture recognition systems have been based on a number of different data sensing technologies and localisations such as off-body, mixed on and off-body and on-body sensing systems.

2.4.1 *Off-body data sensing*

Up until the mid 1990's ACS were almost exclusively based on off-body sensing technology and in particular multi-camera video based systems. This was partly because these off-body systems were quite accurate and partly because cheap, small, accurate sensors that could be used on-body did not exist until then. However, some of the drawbacks of off-body camera systems include that they can be very expensive to acquire; often need to be set up in specialist studios or require very exact placement of the cameras and only operate over relatively short distances.

Multi–camera video based systems generally require either a specialist studio where light conditions can be closely controlled and where camera offsets can be measured in minute detail or if they are outdoor systems then the cameras need to set up in fixed positions so that their offsets can be precisely calculated. Often additional lighting is required specifically for the cameras.

This means that people performing the activities either need to go to the studio to perform them or the place where they normally perform their activities needs to reworked so that it can accommodate the multiple cameras and lighting equipment. This, in turn, means that it becomes much more difficult in those situations to capture real life activities that have not been influenced, often dramatically, by the data capture equipment and/or environment.

The amount of pre-preparation means that it can be difficult in some circumstances to capture spontaneous activities or time-consuming and expensive to capture rare activity. The relatively short range of the camera systems of tens of metres means that some activities that require large areas could never be captured in this manner. These constraints hinder activities that take place across a widely dispersed geography such as cross-country Horse Trials where the field of play can be square kilometres in size or events such as road cycling that may cover hundreds of kilometres. Even with large numbers of cameras, some movements can be occluded and so may be difficult or impossible to capture. Lastly, these systems tend to be relatively expensive to acquire, often costing tens of thousands of dollars to purchase.

These systems can be very accurate despite their drawbacks and were, until recently, considered the Gold Standard and have been used successfully in a large number of research projects. This mature technology can be very effective, especially when basing classification on human skeletal models. A small subset of examples of this research includes Barbic et al. (2004); Brand, Oliver, and Pentland (1997); Campbell and Bobick (1995); Darell and Pentland (1996) and many others. in 1997 Aggarwal and Cai published a review of over 38 projects in this area. Later, Moeslund and Granum (2001) published another comprehensive survey of over 130 papers many of them not part of the earlier review.

Simpler, stereoscopic video systems such as that used in Microsoft's Kinect are also available. Such systems have been used in research such as Biswas and Basu (2011); B. Ni, Wang, and Moulin (2013). While less mature and with usually less range than the multi-camera systems, such systems are significantly cheaper to acquire than the more expensive, multi-camera systems. These single camera sys-

tems otherwise have similar characteristics, both beneficial and non-beneficial, as the multi-camera systems.

2.4.2 *Dual on and off-body data sensing*

Some range-finding/triangulation type systems using broadcast/receive technologies like RFID and RADAR have been used successfully in a limited number of published research projects. Researchers have included Alemdar (2014); Asadzadeh, Kulik, and Tanin (2012); Bannach, Lukowicz, and Amft (2008); Buettner, Prasad, Philipose, and Wetherall (2009) with RFID and Y. Kim and Ling (2009) with RADAR.

These systems tend to be more recent and less popular, perhaps because they are perceived to be more complex to configure or more difficult to miniaturise. While these systems can overcome occlusion problems by having sensors mounted on a participant's body they suffer from some of the same pre-preparation and range issues as the video based systems because they require either the transmitter or the receiver to be located off-body in precisely known positions and within the effective range of the sensing technology.

2.4.3 *On-body data sensing*

On-body sensors have the advantage of not suffering from the occlusion problems faced by video based systems; not requiring extensive preparation or changes to the environment where the activity takes place such as the multi-cameras and specialist lighting of camera systems; of being able to be used in activities such as road cycling and cross-country horse trials that cover extensive distances; are capable of capturing real world activities and are becoming smaller, less obtrusive and are often much cheaper than other systems.

> *Body-worn sensors are well suited to collecting data on activity patterns over extended periods of time. In contrast to other approaches, such as laboratory-based systems or video analysis, they can be used under conditions of free living with minimal inconvenience to the user.* S. J. Preece et al. (2009, p. 2)

However, until quite recently on-body sensors were not considered as accurate as the high–end multi-camera systems; could be bulky to wear and therefore raised issues of affecting how participants moved in some cases and require their own power source which can limit the length of activities monitored to minutes or hours.

A number of different technologies have been used for on-body sensing and some of these include acoustic systems such as in Lester, Choudhury, Kern, Borriello, and Hannaford (2005); J. Ward et al. (2006). Visible light systems such as in Lester et al. (2005). Barometric pressure systems such as in Lester et al. (2005). Multi-sensor combinations with fused inputs such as in Lester et al. (2005); Stiefmeier et al. (2006) and inertial measurement systems using MEMS accelerometers, gyroscopes and magnetometers.

The concentration of ACS based on off-body camera systems really started to change when the MEMS accelerometer began to appear. Walter (2007), in his "*The History of the Accelerometer*", reports that the Automotive industry's requirement for a low-cost, highly reliable accelerometer to trigger Air Bags lead directly to the introduction of the Analogue Device's MEMS based ADXL50 accelerometer that was introduced in 1991 and which went into high-volume manufacturing in 1993. The ready availability of this small, low-cost, low-power, highly reliable sensor changed the research environment and these sensors were soon picked up by researchers with a need to have an on-body device to monitor human movement.

During the mid to late 1990's research started to appear that was based on the use of these inertial sensors. Early work based on on-body inertial sensors includes Patterson et al. (1993), this seems to have been one of the earliest publications that classified activities using accelerometers. These researchers used an Actigraph accelerometer to differentiate daily activities such as walking, running, stair climbing, knee bends, reading, typing, playing video games, and performing a mental arithmetic task. Veltink et al. (1996), another early work, classified standing, sitting, lying, walking, ascending stairs, descending stairs, cycling; separated activities into a static (standing, sitting and lying) and dynamic class (walking, ascending and descending stairs and cycling) with both high and low-pass filters, then classified cyclic activities based on signal mean and standard deviation.

Verplaetse (1996), went further with their sensor and used an IMU using an IMU consisting of an accelerometer, gyroscope and other sensors to sense motion, rather than classifying activities. Foxlin, Harrington, and Pfeifer (1998), reported on a wireless motion tracking system for Virtual Reality (VR) using accelerometers and other sensors. Other on-body inertial sensing included Bachmann et al. (1999), who reported on orientation tracking for humans and robots using accelerometers and other sensors; Foerster, Smeja, and Fahrenberg (1999), who detected human posture and motion using accelerometers; Perng, Fisher, Hollar, and Pister (1999), who describe a gesture sensing glove that used multiple accelerometers; Randell and Muller (2000), who reported on a system that used a single accelerometer to detect "*walking, running, sitting, walking upstairs, downstairs and standing*", using RMS and integrated values from the X and Y axis of the accelerometer as the only features. These researchers segmented their data into windows 2 seconds wide and their classification was person and clothing specific; Van Laerhoven and Cakmakci (2000), who described a pair of pants with attached accelerometers and an associated system that allows end users to "teach" the pants which activities to highlight; J. Lee and Ha (2001), who used multiple on-body accelerometers to capture human motion in real-time and Mantyjarvi, Himberg, and Seppanen (2001), who used a 256 sample sliding window to segment data from two accelerometers that were hip mounted in order to classify "*walking in a corridor, walking down the stairs, walking up the stairs and opening doors*". These researchers used a Wavelet transform to process the segmented data and then used Principal Component Analysis (PCA) and Independent Component Analysis (ICA) to extract features from the processed data.

These early, pioneering studies were then followed by a flood of other work. Today, on-body inertial data sensing commonly sits alongside the highly accurate multi-

camera based systems and the cheaper Kinect type camera systems as the most popular techniques of data collection for movement sensing and activity classification. In some areas of research other sensing technologies are also used to supplement inertial sensors and in some cases these alternate, non-inertial on-body sensors replace the inertial sensors but the majority of on-body sensors used for activity classification are now inertial ones.

The author has recognised the benefits of using these small, cheap, relatively low powered devices in the same way as many other researchers have and the data used in this research was all captured using an inertial sensor that included a MEMS accelerometer, a gyroscope and a magnetometer.

2.5 ALTERNATE USES FOR INERTIAL DATA

Having established that on-body, inertial sensors are used by a number of researchers to capture data, it is useful to look briefly at the broad range of uses that this data is being used for. Three main threads are seen in the current research, these are motion capture, gesture recognition and activity classification.

2.5.1 *Movement Identification or Motion Capture*

When used in this area, generally researchers use multiple sensors, often placed close to body pivot points such as lower arm, wrist, just below shoulder, waist, head, thigh and similar places. When used in this manner, researchers such as Bachmann (2000); Brodie, Walmsley, and Page (2008); Brunetti, Moreno, Ruiz, Rocon, and Pons (2006); J. Lee and Ha (2001); Mayagoitia, Nene, and Veltink (2002); Zhu and Zhou (2004) have fused the inputs and used a kinematic model of the human body to estimate posture and movement. Generally, when used in this manner researchers utilise fused data from multiple, complementary inertial sensors (accelerometer, magnetometer & gyroscope) with continuous error correction, often using some form of Kalman filter, or in some cases specialist continuous error correction filters such as proposed by Madgwick (2010) to deal with noisy and skewed data from the sensors.

Other researchers such as Y. Kim and Ling (2009); H. Zhou and Hu (2005); H. Zhou, Stone, Hu, and Harris (2008) (arms) and Lang, Kusej, Pinz, and Brasseur (2002) (head) have concentrated on a sub-set of the body while using similar data fusion and error correction techniques. Roetenberg, Slycke, and Veltink (2007) is an example of other researchers who have slight variations on this theme and in their work they use a novel sensor in conjunction with an accelerometer and gyroscope to aid in body posture tracking.

In some ways, these uses are similar to the multi-camera based motion capture systems. Often these systems make use of a ”suit” with fixed pockets for the sensors so that the sensors are placed in known positions with known orientations and the sensors can be hard-wired back to a logging/processing device. A number of commercial systems are now available that use this technology.

Yet other researchers such as Bernmark and Wiktorin (2002); Mizell and Cray (2003) have reported on work that uses a single inertial sensor to estimate orientation

and simple, non-kinematic movement. This work naturally leads into the next main thread, that of gesture recognition. For a more in-depth review of other work in the area of inertial motion capture and movement identification the author recommends Welch and Foxlin (2002) or more recently H. Zhou and Hu (2008).

2.5.2 *Gesture Recognition*

Gesture recognition systems use inertial data to recognise short, simple system standardised and usually artificial movements. Often these gestures are then used to control some other process, much like a mouse is used on a windowed desktop computing system to control the desktop computer operating system and applications. Often these projects use a Quantize, Model and Classify pipeline with HMMs popular as a Model. Researchers in this area have included Amft, Amstutz, Smailagic, Siewiorek, and Troster (2009); A. Benbasat and Paradiso (2002); A. Y. Benbasat (2000); Madgwick (2010); Mantyjarvi et al. (2001); Mantyjarvi, Kela, Korpipaa, and Kallio (2004); Ramamoorthy, Vaswani, Chaudhury, and Banerjee (2003); Schlomer, Poppinga, Henze, and Boll (2008); Westeyn, Brashear, Atrash, and Starner (2003). Other researchers who have not used HMMs include Fels and Hinton (1993) who used five neural networks, trained via back propagation to implement a hand-gesture to speech system. H. Lee and Kim (1998) use a two stage process with a threshold model as the first step to isolate gestures and then apply a HMM to classify these isolates into standardised gestures. Some researchers have used additional sensors to augment their gesture recognition systems. For example, Ogris, Stiefmeier, Junker, Lukowicz, and Troster (2005) fuse ultrasound data with inertial sensor data to improve their recognition rates.

A subset of gesture recognition research deals with the idea of motion primitives that can be combined either to form more complex gestures or activities and are analogous to the idea of letters of an alphabet that make up words and concepts. These motion primitives are a very similar idea to the "Movements" described in the next section. As an example, Amft, Junker, and Troster (2005) describe their work recognising arm gestures that are related to eating and drinking activity. Minnen, Starner, Essa, and Isbell (2006) and Vahdatpour, Amini, and Sarrafzadeh (2009) take a different tack by attempting to discover sparse motifs ("*sets of similar sub-sequences*") within the sensor data using a three stage, bottom-up approach.

Another subset includes semi-natural gestures that are used to annotate or label data. This includes Chambers, Venkatesh, and West (2004); Chambers, Venkatesh, West, and Bui (2002); Roh, Christmas, Kittler, and Lee (2008) where cricket umpire like gestures were used to annotate a video of a cricket game.

Yet other subsets include researchers who use static poses as gestures (Perng et al., 1999), motion trajectories (Psarrou, Gong, & Walter, 2002), turning points of arm movements (Zinnen & Schiele, 2008) and location, angle and trajectory (Yoon, Soh, Bae, & Yang, 2001). T. Ni (2011) contains an extensive review of gesture recognition systems and would be a good starting point if the reader requires a more in-depth review of this area.

2.6 SUMMARY

This chapter defined the activities of interest and introduced any reader without a background in horse riding to some commonly held ideas relating to Mounting and Dismounting. Then the chapter introduced the bigger picture of a possible wearable coach for horse riders based on data from on-body inertial sensors. Within the bigger picture, the chapter looked at some ideas associated with Adaptive Systems, context as an aid to classification and context constrained systems, how successful classification can assist with the possible inference of next activities and some additional ideas that intersect wearable coaching systems and activity classifiers. Then given the interest in activity classification the chapter takes a quick look at off-body versus on-body sensing and some associated uses for inertial data collected from on-body sensors.

The following chapter will review the literature surrounding inertial based activity classification systems to highlight possible gaps and to point to techniques that may be appropriate to possibly filling those gaps.

Chapter 3

LITERATURE REVIEW

The goal of this literature review is to review relevant research literature to establish a suitable technique and/or approach that can then be used to (iteratively) develop the artefact of interest, an activity classifier that will successfully classify the Punctual activity Mounting from within real world horse riding activities.

3.1 INTRODUCTION

Towards the end of the previous chapter two alternate uses of on-body captured inertial data were discussed with brief reviews of current research in those areas. In this chapter the third main use of on-body inertial data, Human activity classification, is reviewed in some detail with the intent of highlighting gaps that are relevant to the activities of interest.

Following on from this is a review of segmented versus unsegmented inertial sensor data within ACS research, arguing that current ACS usually segment their input data into ”windows” and classify Durative or cyclic activities while mostly ignoring unsegmented input data and Punctual activities. It is also noted that those Punctual activity classifiers that have been reported on, that utilise inertial data from on-body sensors, have used segmented input data for classification.

RC techniques are briefly described along with an explanation of why they are a useful tool for classifying un-windowed spatio-temporal data. Lastly the author concludes that ACS research on Punctual activities using un-windowed data has novelty and this is used to justify the proposal to construct a RC based, Punctual activity classifier that processes unsegmented, unscripted real data from the domain of horse riding.

This literature review is split into two parts. The first part consists of the literature review that was done prior to the development of the classifier artefact for classifying Mounts from continuous, unsegmented inertial data captured from real life, unscripted riding sessions. This part of the literature review drove the iterative development of this artefact and in that sense drove the research. This part of the literature review starts below with section 3.2.

The iterative development of the classifier artefact took place over several years and during that time inertial based activity classification research by other researchers continued. The second part of the literature review then, at the end of the iterative development process reviewed some of the relevant research literature that happened

in parallel with the development from this research and this additional research literature is then contrasted with both the initial literature review and and with the insights that developed out of the iterative development within this work. this second part of the literature review starts with section 3.3 on page 58.

3.2 INITIAL LITERATURE REVIEW

Activities are seen by researchers such as Bobick (1997) as sitting between Movement and Action and could be loosely described as behaviour of a particular or purposeful kind. In this context a **Movement** might be "*raising the right leg 15cm off the ground*"; an associated **Activity** might be a series of leg and body movements that are called "*walking*"; and an associated **Action** might be "*walking from home to the bus stop*".

It is worth noting an alternate **Activity** associated with this series as being a series of leg and body movements that could be called a "*step*" and in this case it can be seen that walking is often composed of a series of alternating steps. It is useful to highlight the relationship between stepping and walking here as this leads into subsequent sections which differentiates different classes of activity.

The next section reviews inertial based activity classification in a general sense and discusses what may have been possible motivations for prior inertial based activity classification as these motivations will have influenced what techniques other researchers have used in this arena. Following on from this, the review highlights that an associated research area, video data based activity classification, has distinguished between Punctual and Durative activities whereas the inertial based researchers have not acknowledged this differentiation. Subsequent sections of this review highlight the lack of differentiation between these classes of activity type as a possible gap in the inertial based research and suggest a possible technique that may be used on inertial data based Punctual activities.

3.2.1 *Motivation for Inertial Data Based Activity Classification*

Much of the initial work using inertial sensors for activity classification including work such as J. Bussmann, Veltink, Martens, and Stam (1994); Foerster et al. (1999); Veltink et al. (1996) came from researchers with a health background who had an interest in how active a person (patient) was overall. These researchers were interested in, for example, if that person walked for ten minutes today versus (perhaps) five minutes yesterday.

These early researchers set out to classify (recognise) relatively common daily activities such as walking, running, standing, lying down, ascending and descending stairs and cycling. This set of activities are performed over an extended time and they each have a repetitive cycle that consists of repeating some sub-activity. For example, walking consists of repetitively taking alternate walk-steps; running consists of repetitively taking alternate run-steps and so on. Even lying down can be considered to be composed of repeating multiple short lie-periods.

The motivation for these early health researchers was to create a tool that they could use to measure how active a person was on a day to day basis so that the data could be used for comparative purposes. As such they were most interested in activities such as walking that generally occur over a time period of minutes or hours versus activities that may only take one or more seconds to complete. As a result, almost all of the activities that the early Health researchers were interested in were Durative activities and so the techniques that they chose for their classifiers were techniques suited for Durative activities.

The first appearance of a non-cyclic (Punctual) activity classifier using inertial data from researchers with a health focus started to appear around 2003 when researchers investigated classifiers for falls. Degen, Jaeckel, Rufer, and Wyss (2003) and Hwang, Kang, Jang, and Kim (2004); Mathie, Celler, Lovell, and Coster (2004) reported on work that used a threshold test to classify falls. In the case of Falls, researchers were focussed on a single event and while they chose different, more appropriate techniques for classifying falls than prior researchers had used for Durative activities these Fall researchers failed to generalise the differences in timing length and the non-repetitive nature of Falls and so an opportunity was missed to differentiate Punctual from Durative activities.

Around the late nineties on–body, inertial data, activity classification research started to appear from researchers with a computer science and/or algorithmic background, I call these the Behaviourists. This research thread includes such pioneers as Foerster et al. (1999), Randell and Muller (2000), Van Laerhoven and Cakmakci (2000), Mantyjarvi et al. (2001) and S. Lee and Mase (2002). These researchers were motivated to improve on the work done by the earlier researchers and so these works followed the same basic techniques introduced by the initial health focussed researchers and concentrated on classifying cyclic "daily activities" type activity. This new research built on, extended and refined the earlier work and a standardised set of techniques started to emerge. This standardised set of techniques was well suited to the Durative activities that these Behaviourist researchers worked with and it is best expressed in a quote from S. J. Preece et al. (2009, p. 3):

> *Most approaches to activity classification, using body-worn sensors, involve a multi-stage process. Firstly, the sensor signal is divided into a number of small time segments, referred to as windows, each of which is considered sequentially. For each window, one or more features are derived to characterize the signal. These features are then used as input to a classification algorithm which associates each window with an activity.*

This process of chopping the input signal up into discrete (usually overlapping) windows is well suited to activities that extend over a reasonably long period of time and in situations when the exact start and end point of each activity is not needed. This is ideal for Durative activities but is often not ideal for very short, Punctual activities where missing the start (or end) of an activity may result in missing it altogether.

Not long after this, researchers with a sports background started to take an interest in activity classification based on data from on-body inertial sensors. The motivation

for some of these researchers was to record different training activities and to record data associated with energy expenditure. One of the earliest works within this thread, Heinz, Kunze, Gruber, Bannach, and Lukowicz (2003), looked at non-cyclic activities (specific Kung-Fu moves) and had moderate success using the standardised set of techniques with scripted, laboratory based activity from two subjects. One of the side effects associated with scripted, laboratory based activity data recording is that the data is often unnaturally consistent due to following the script and where multiple attempts are made to record the same activity over a limited time period then the repetition of the activities introduces a repetitive element to the data that is recorded even when the activities are not inherently repetitive.

The issues around the unnaturally consistent data associated with scripted, laboratory based activity classification are now well recognised and has been reported by a number of researchers in the field such as Foerster et al. (1999), Bao and Intille (2004) and Ravi, Dandekar, Mysore, and Littman (2005) who reported high classification accuracy (84%–95%) from scripted, laboratory based data versus much lower accuracy (24%–66%) when using the same techniques with real world, unscripted activities.

Sports focussed, on-body, inertial data based activity classification research has been sparse since Heinz, Kunze, Gruber, et al. (2003) with Michahelles and Schiele in 2005 and Harding, Mackintosh, Hahn, and James in 2008 both reporting on work that classified cyclic sports activities using the standardised set of techniques. Leaving the author's own work aside, the next reported work is Mitchell et al. (2013) where the authors classify five cyclic (Durative) and two non-cyclic (Punctual) soccer and field hockey activities using the standardised set of techniques. While Mitchell et al. (2013) included both Durative and Punctual activities within their work they did not differentiate between the two classes of activity and used the same technique for all activities. Their reported results showed high classification accuracy for 4 of the 5 Durative activities and very low classification accuracy for the 2 Punctual activities. The fifth Durative activity, dribbling a ball was difficult to differentiate from walking and running, two of the other Durative activities. This reinforces a difference between the two classes of activity but in this case the researchers did not pick up on the cause of these differences.

From 2004 and onwards, on-body, inertial data based activity classification work started to appear from researchers with what I categorise as an Auditing focus. The motivation for these researchers was to record activities, often in a work situation, to ensure that required activities were done. An example of this would be recording a worker screwing in a particular screw to make sure that all screws were installed. In the case of the 2004 work, (Lukowicz et al., 2004), the researchers were working on workshop activity recognition that included both Durative and Punctual activities.

By this time the so called "standardised" way of cutting the input data up into discrete windows was so ingrained that again, this work plus Stiefmeier et al. (2006) [both Durative and Punctual bicycle repair activities] and Marin-Perianu, Lombriser, Amft, Havinga, and Traster (2008) [both Durative and Punctual automotive assembly activities] followed the established way of segmenting the signal into windows, calculating features from those windows and then classifying the activities based on

those calculated features. In all three of these works the Punctual activities were a small minority of the total activities audited and so all researchers achieved moderately good results using the standardised techniques. Where these researchers found that their Punctual activities were not responding well they preferred to add additional sensors (and therefore additional information) such as proximity sensors rather than differentiating the Punctual activities and using a different classification technique with those activities.

However, in Zappi et al. (2008) [both Punctual and Durative automotive assembly activities, mostly Punctual], in some cases, the signal is not segmented and instead each raw sensor sample is instead compared with two threshold values, (somewhat reminiscent of the fall classification research from the health focussed researchers), and converted into one of three values that are then input into a HMM for initial classification. While the researchers do not state which of the various activities they use the non-standardised, non-segmented data technique with, it is highly possible that they use this technique to improve the classification of the Punctual activities of interest because all but one of the ten activities of interest is a Punctual activity. In any case, it is interesting to note the use of a set of techniques that does not fit that which other researchers have considered a standard, even though, once again, this has not been generalised into Durative and Punctual classes by the researchers themselves and it is also interesting to note the use of the HMM for classification acknowledges an understanding that for some activities, the temporal order of the features used for classification are important. This acknowledgement of the usefulness of temporal order is generally not seen with other inertial data based activity classification researchers, particularly those working with Durative activities.

Almost all of the on-body, inertial data based activity classification research can be grouped into the four motivational threads discussed above, namely Audit focussed, Behaviour (computer science and/or algorithmic) focussed, health focussed and Sports focussed. Some other minor threads do exist such as Military focussed (Minnen, Westeyn, Ashbrook, Presti, & Starner, 2007) and Security focussed (Yin, Yang, & Pan, 2008), but these minor threads do not or have not differed from the Health and Behaviour focussed work and there is not enough of it to warrant detailed analysis or reporting.

3.2.2 *Differentiating Inertial Based Activity Classifiers*

Earlier, in the Introductory chapter (see page 2) we offered definitions of Punctual and Durative activities. We also noted that Niebles et al. (2010) offered a very similar definition to our own but from the perspective of Human activity classification using video data. This definition is also supported by Amer, Xie, Zhao, Todorovic, and Zhu (2012), also from the video data based research community, who use "*punctual*" to differentiate activities with no repeating elements from "*repetitive*" activities. See Amer et al. (2012, p. 2 last para), Amer et al. (2012, p. 3 first para) and Amer et al. (2012, p. 4 first para).

H. Zhang, Zhou, and Parker (2014) (also from the video data research community) use "*punctual*" in the same manner as Amer et al. (2012), to differentiate non

repeating from repetitive activities, see (H. Zhang et al., 2014, p. 2, Section 2, para 2). In addition, H. Zhang et al. (2014) point out that trying to distinguish activities from continuous data, as opposed to data that has been preprocessed to highlight and extract particular activity sequences is a difficult task as it is difficult to find the exact point where one activity finishes and another activity starts. This issue is of particular significance for Punctual activities because their non-repetitive nature means that they are usually quite short whereas most repetitive activities are much longer. It is much easier to miss the shorter activities.

At the top of page 393 of Niebles et al., we note that the authors state that

> Durative and Punctual activity classifiers require different classification algorithms in order to take advantage of the different properties of each type of activity and that as a result, classification algorithms designed for one class of activity perform poorly on the other.

Our own review of the relevant literature shows that within the realm of inertial based activity the lack of differentiation between Durative and Punctual activities (until now) has muddied the discussion. Nevertheless, researchers within inertial based activity classification have tended to use different algorithms and approaches for the two classes of activity and where they have tried to use the same approach across both then they have have been largely unsuccessful when classifying the activity class that is not the primary focus of the algorithm used. Following, we differentiate and discuss the inertial based activity classification with this focus.

3.2.3 *Inertial Based Durative Activity Classification*

The daily activities that were a focus of both the early Health orientated and Behavioural orientated researchers such as walking, running, standing, ascending stairs, descending stairs and the like are Durative activities as they relate to continuing, cyclic action. (Smith, 1999). The common classification approach is to use the Quantize, Model and Classify pipeline e.g. (Bao & Intille, 2004) who used Fast Fourier Transform (FFT), a predefined feature set and a Decision Tree for classification. According to S. J. Preece et al. (2009, pp. 3), in most cases the sensor signal is divided into small time segments and these are considered sequentially (Quantise). Features are calculated/derived from each segment so as to characterise the signal (Model), and then the features are used as input into a classification algorithm (Classify).

S. Preece, Goulermas, Kenney, and Howard (2009, pp. 1) suggests that three main techniques are used to derive features from the segmented sensor signal, these are **Direct** features such as segmented signal mean, standard deviation, variance, Root Mean Square, median, skew, kurtosis, 25% percentile, 75% percentile, Mean Absolute Deviation and Zero Crossing Rate; **Frequency** features such as segmented signal FFT, frequency domain entropy, spectral centroid, spectral spread, estimation of frequency peak, estimation of power of the frequency peak and signal power in different frequency bands; and **Wavelet Analysis** features calculated from various forms of Discrete Wavelet Transform (DWT) decomposition.

Looking at these activities through a temporal lens it can be seen that they all have a start phase, a repetitive central portion and an end phase and looking at walking, for example, it could be said that a person starts walking, then they are walking and finally they will stop walking at some point.

However, those early researchers were by-and-large uninterested in the start and end phases of Durative activities as they were interested in quantifying the general extent of an activity as opposed to finding the exact start or end of its instantiations and so classifiers for those Durative activities responded well to the data being segmented into fixed windows and then having statistical and frequency features extracted from the sensor signals within these windows. For these researchers not identifying a couple of seconds of activity at the start and end were inconsequential compared with correctly identifying the minutes or hours of activity between the start and end.

The author concurs that these techniques are indeed appropriate for Durative activities but to only consider techniques that work well for these cyclic activities where temporal order has little relevance is, as Niebles et al. (2010) has stated, misses the opportunity to use differing and potentially more appropriate techniques for Punctual activities.

3.2.4 *Inertial Based Punctual Activity Classification*

Some activities, such as picking up a cup, opening a door and mounting a horse (as examples), either only have a single stage or consist of non-repetitive stages and are often very short. These Punctual activities do not contain a repetitive cycle and their relatively short duration mean that correctly identifying the start and end phase can be very important. In addition, many Punctual activities are of variable length e.g. mounting a horse (using the original, extended definition of mounting) took an average 5.48 seconds with a standard deviation of 2.28 seconds (Hunt, 2009, pp. 153). This temporal variability makes it difficult to isolate the start and end of a Punctual activity when the input data is cut up into windowed segments as it becomes difficult to segment the input data in a way that ensures that the start, middle and end stages are consistently captured without risking the activity signal being overwhelmed by non- activity data. In this area Junker, Lukowicz, and Troster (2004, pp. 1) states:

> "With a scheme that partitions e.g. the signal into segments of predefined length, it is very likely to miss the exact beginning and end of the relevant movements which is critical particularly for activities of short duration".

In addition, spectral type quantization techniques that are often successfully used on cyclic Durative activities are unlikely to be successful with Punctual activities as, by definition, Punctual activities are not cyclic in nature.

One activity that early health-focussed researchers were interested in that did not fit the Durative activity mould was "Falls" and it is in this area that differing techniques being used to classify this activity are first seen. Degen et al. (2003), Hwang et al. (2004) and W. H. Wu et al. (2008) for example, segmented their sensor data and then applied a simple threshold test to an average of each data segment to distinguish falls from non-falls; T. Zhang, Wang, Xu, and Liu (2006) segment their data and

then look for an "acute change" from the prior segment; Nyan, Tay, and Murugasu (2008) on the other hand do not segment their data and instead calculate approximate thigh and waist orientation and rotational velocity, they then distinguish a fall when certain orientation and/or rotational velocity thresholds are exceeded. Doukas and Maglogiannis (2008) combine an acoustic sensor (microphone) with an accelerometer on the patient's foot to detect falls, interestingly, they seem to use unsegmented accelerometer data alongside sound peak frequency and amplitude data in their classification process but force a fixed, five second, post-classification segmentation window on the primary classification results to provide a secondary and more accurate classification. They use a similar technique in their later, related work (Doukas & Maglogiannis, 2011) but also add in off-body cameras and off-body microphones.

All these above works target "falls" as the primary activity to be classified, some health related activity classification research includes falls alongside other more traditional activities. When falls are only one of a number of activities to be classified, most researchers use traditional classification techniques.

Over time other researchers with different backgrounds and world views started to become interested in sets of activities that included Punctual activities. Heinz, Kunze, Gruber, et al. (2003) report on some early work within a sports context when they classified a number of Punctual and Durative activities from the sport of Kung Fu. Their approach was to use unsegmented raw gyroscope data from a single axis from a neck mounted sensor to classify "turns", while they used more traditional approaches for their Durative activities. While the authors of this work used a different technique to classify the Punctual activity (turns) they did not define or distinguish any fundamental difference between Punctual or Durative activities and simply called them all "activities". Nevertheless, their use of unsegmented data in this manner supports the contention that some activities (Punctual ones) are not best classified using the standardised techniques developed for Durative activities with a cyclic nature.

The Auditor focussed researchers included Punctual activities within their activity sets from the beginning and work such as Lukowicz et al. (2004) where the authors classify activities encountered within a woodwork workshop demonstrates this. Similar audit focussed work has included Amft, Lombriser, Stiefmeier, and Troster (2007); Junker, Amft, Lukowicz, and Troster (2008); Stiefmeier et al. (2006); J. Ward et al. (2006). Again, the authors do not differentiate between Punctual and Durative activities and use the same traditional techniques to classify all activities. As an example, J. Ward et al. (2006) use segmented data, Mean and Crossing Count as features but acknowledged the importance of the temporal aspects of the signal by using a HMM as a classifier for all activities, both Punctual and Durative, during their audit orientated study.

Other audit focussed work based on instrumented "smart houses" such as Logan, Healey, Philipose, Tapia, and Intille (2007) used hundreds of sensors including a small number of wearable sensors and set out to classify large numbers of activities however, as reported, all except three activities took at least a minute to do (Logan et al., 2007, pp. 10) and most of the described activities were Durative in nature. With so few Punctual activities the authors followed a traditional approach of segmenting the large number of sensor readings, calculating features from each segment and then

classifying activities based on those features. However, the authors did recognise that this was a less useful process when dealing with the shorter (more likely to be Punctual) activities. Once again, this supports the contention that Punctual activities may well be better served by techniques that cater for their shorter length, lack of a cyclic pattern and in many cases the significance of the temporal order of the signal features.

Other authors have kept the more traditional classification approaches developed for Durative activities when classifying Punctual ones but recognising that classification rates for Punctual activities has been less reliable in these cases have then turned to other techniques to try to improve reliability. In a number of cases the researchers have added additional sensors in their quest for reliability. One series of studies out of the Wearable Computing Lab at ETH in Switzerland based around motor vehicle assembly tasks that included Punctual activities, used off-body proximity sensors, ultrasonic sensors and other off-body data to complement the inertial data when they achieved reasonable reliability with Punctual activities. Specific research examples from this series include Stiefmeier et al. (2006), where input data was manually pre-segmented and supplemented with ultrasonic sensors; Zappi et al. (2007), where input data was pre-segmented and acceleration features were extracted; Lombriser, Bharatula, Roggen, and Troster (2007), where input data was segmented using a sliding window protocol and supplemented with sound data from a microphone and ambient light data from a visible light sensor; Stiefmeier, Roggen, Ogris, Lukowicz, and Troster (2008), where input data was pre-segmented into interesting areas and supplemented with proximity sensors on the motor vehicle, force sensors on the participant's body and RFID tags on the tools. These researchers have recognised that the standard techniques using standardised data was insufficient but in their cases they have preferred to add additional information (from the extra sensors) rather than trying different classification techniques. This is entirely acceptable but misses an opportunity to widen the range of techniques available to inertial data activity classification researchers.

Koskimaki, Huikari, Siirtola, Laurinen, and Roning (2009) report on a similar study to the ETH studies where data was recorded from a fixed position (a positional domain constraint) that represented a particular "post" in an assembly line. They segmented their input data using sliding window, then extracted features from the widowed data. Both series of assembly-line studies are similar in many ways to the laboratory data captured using a wooden "horse" in a fixed position for this research.

Another ETH study by Junker et al. (2008) looked at gestures (Punctual activities) from everyday life with some success and in this case the input data was pre-segmented into sections likely to contain activities of interest using a sliding window technique, they used data from multiple inertial sensors and then extracted features from the segmented inertial data.

J. Ward et al. (2006) used an on-body non-inertial sensor to improve reliability and still reported significantly less reliability for the Punctual activities ("*Turn screwdriver*" and "*Pick up file from drawer*") compared with the Durative activities in their study.

Most of the Punctual classification studies used explicitly Constrained Domains (such as specific assembly tasks for production line systems) or implicitly Constrained Domains such as a specific door when classifying opening a door, or light switch when turning on a light (Junker et al., 2008) to reduce the search space and simplify the task.

There has not been a prevalent understanding of differences between Punctual and Durative activities within the published inertial based activity classification research. Where researchers have focussed on an activity or activities that can be classified as Punctual, such as falls, then the resultant classifiers are often non-traditional in their techniques but where research has included a minority of Punctual activities versus Durative activities then researchers have tended to classify all activities using the same technique and that technique has usually been a traditional one even though some researchers have acknowledged that traditional techniques have been less successful with shorter duration (often Punctual) activities.

One potential solution for finding short sequences of variable length is to divide the search into two parts where the first part of the search uses any number of techniques to find a "similar" signal sequence and then the second part of the search uses a different technique to search the area around the identified sequence for a more exact match that more clearly identifies the start, middle and end sequences. For example, Junker et al. (2008) uses this technique by utilising a similarity search at first and a HMM as their secondary classification technique in their work classifying activities within industrial assembly tasks. While this two staged technique works it has an obvious drawback, especially when attempting to do time critical activity classification, as the technique requires two passes over the same data.

Other techniques that researchers have used on unsegmented data to classify Punctual activities have included threshold limits for falls as in Bourke, OBrien, and Lyons (2007), Bourke, ODonovan, and OLaighin (2008) and Nyan et al. (2008) or using multiple thresholds to codify the data and then classifying the codes (Zappi et al., 2008). Minnen et al. (2006) use a much more sophisticated variation of codifying the data but their technique does segment the data, although the technique calculates the segment window size rather than using a fixed segment window. Minnen et al. (2006) also use a two stage process where HMM are used to classify the activities during the second step.

A very small minority of researchers have used unsegmented data to classify Durative activities, J. B. J. Bussmann, Veltink, Koelma, Van Lummel, and Stam (1995) used unsegmented data to differentiate static activities such as standing and lying from dynamic activities such as walking and S. H. Lee, Park, Hong, Lee, and Kim (2003) do something very similar.

3.2.5 *Complex Activities*

A third class of activities may be called Complex activities and are typically composed of a series of Durative and/or Punctual activities. Complex activities include activities such as eating, cooking, riding, fishing. In general, researchers have not attempted to classify these complex activities and where they have such as in Logan et

al. (2007), where the authors tried to classify "eating" as one of a number of everyday activities within an instrumented home, they had only moderate success at best.

There has been limited work done that focussed on classifying complex activities, perhaps in recognition of the difficulties in this area. Marin-Perianu et al. (2008) is one reported work that has attempted complex activity classification but used data from off-body sensors with Fuzzy Inference with state change as their classification engine.

3.2.6 *A Wider Inertial based Activity Classification Review*

What has been missing from the inertial data based activity classification research dialogue has been a clear differentiation between Durative and Punctual activities and an acknowledgement that techniques that are successful in one area may not be ideal in the other area.

This work provides a definition that distinguishes Punctual activities from Durative activities based on differences of the length of completion time for activities and, perhaps more significantly, on differences in the rhythmic/cyclic nature of Durative activities versus the non-rhythmic/non-cyclic nature of Punctual activities. In general video data based activity classification researchers have been reasonably clear about distinguishing Punctual and Durative activities (see Niebles et al. (2010), Amer et al. (2012) and H. Zhang et al. (2014)) however researchers who do activity classification based on inertial data have usually failed to differentiate Punctual, non-cyclic human activities from Durative, cyclic activities and where they have differentiated these different classes of activity their definitions have not been particularly clear or consistent.

Some inertial data based activity classification researchers have called Punctual activities non-periodic activities (Junker et al., 2004) or gestures (Chambers et al., 2002)in order to differentiate them from (real) activities while others have used terms such as short activities or very short activities to try to differentiate them from longer activities(Ravi et al., 2005). Almost none of these attempts at differentiation have been either well defined nor have they demonstrated any understanding of the differences in the rhythm or frequency of Punctual versus Durative activities and so while this research is not the first publication to define and differentiate Punctual and Durative activities, this work uses these prior definitions to highlight the general lack of research into the area of inertial data based Punctual activity classification versus inertial data based Durative activity classification.

To be clear, it is not being suggested that techniques that have been successfully developed for classifying Durative activities are not applicable to Punctual activities and vice versa, but rather that by not recognising the differences between the fundamental characteristics of Durative and Punctual activities may lead into using techniques that are optimal for most cases of a particular class and fail to consider alternate techniques that may be optimal for the alternate class of activities. This seems to be particularly an issue when classifying Punctual activities as the following analysis shows. In order to back up the contention that there are differences between Durative and Punctual activities and the techniques used to classify them

the author undertook a short review of publications related to inertial based, activity classification research. This review is described in the following sections.

3.2.7 *Initial Review methodology*

The methodology included:

1. The use of widely cited reviews or key publications plus the author's own searches via Google Scholar to find publications
2. To include only inertial based activity classification publications
3. To exclude gesture recognition and motion tracking work
4. Where the same or very similar authors re-publish the work under differing titles then only one, representative publication, has been included. However, new work from research using the same data has been included where it is deemed different enough. Of course, decisions to include or exclude work is based on interpretations and is open to challenge
5. Where a publication is cited in more than one source then it is only included in the review once
6. The author believes that while the list of reviewed work is not complete that it is comprehensive enough to allow valid conclusions to be drawn

The list of surveyed publications includes 137 publications from 1995 to early 2014. The reviews and key publications that formed the core of works from which citations are drawn are: Bao and Intille (2004); Baca, Dabnichki, Heller, and Kornfeind (2009); Companjen (2009); S. J. Preece et al. (2009); Avci et al. (2010); Brush, Krumm, and Scott (2010); Plötz (2010) and Lara and Labrador (2013). The individual works are not referenced directly within this section but are all listed within the references chapter. Works used within this review are noted with an asterisk in the references chapter. Where specific characteristics of a particular publication are commented on then that work is directly referenced in this section.

The works considered within this review are characterised by a number of different attributes, these attributes are:

Author - The author(s) of the publication that was reviewed

Year - The year that the reviewed publication was published

Inertial - Confirmation that the publication had used inertial data during classification; note that this does not preclude the use of non-inertial data in addition to the inertial data

Activities - A short description of the activities that were researched within the publication; e.g. daily activities, bicycle repair activities, falls, office tasks, juggling, workshop activities, Kung Fu moves, exercises, workshop activities, cycling, walking and transitions ...

Placement – Where the sensor(s) were placed on the subjects bodies

Segmentation – Was the inertial data segmented and features derived from the segments prior to classification versus using the unsegmented inertial sensor data for direct classification. In some cases this was implied where the authors of a work did not specifically say that their data was segmented but where they used features calculated from their data such as means, medians, standard deviations, minimums, maximums, ranges and frequency transforms such as Fourier transforms. Where the author decided that segmentation was implied, it is noted as implied.

Domain – The (implied) primary domain or background for the researchers who published the works; e.g. Health, Audit, Sports, Security or Behaviour; this was often assumed from where the work was published (medical versus computational versus sports conferences & journals for example) or from the emphasis within the publication (e.g. recording the total period of activities in order to estimate metabolic effort implied a health view point)

Class – Durative, Punctual or both. This was derived from the author's own judgement.

Situation – Laboratory or Real World situation; note that the author interprets "laboratory" in its widest sense. For example, a study done along a university corridor and staircase is labelled as a laboratory as it is a staged situation with constraints on choice of environment or interactions

Scripted – A note of if the activities that subjects performed were scripted or completely free-form. Note that where researchers reported that their subjects were given a list of activities to perform but were not told how to perform them then that was generally classified as scripted activities especially when performed within a laboratory environment. However, where subjects performed unprompted activities within a realistic free-living, laboratory environment then that was classified as unscripted, even though they were done within an artificial "laboratory".

Numbers – The number of participants who performed the activities during the data capture part of each piece of research

In addition a note was kept of the source of each publication (where the citation was found if a review was used) and a general notes field for any particularly noteworthy point within the publication.

3.2.8 *Activity Types*

A plot of the frequency of various activity types encountered during the review is shown in Figure 3.1 on page 40 and attention is drawn to this plot during this part of the discussion.

Within the activity type, the single largest common description was “Daily activities”, at 64.23% of all work. This description was somewhat ill-defined and seemed to include and exclude activities at the author’s whim. However, this description generally included walking, running, standing, stair climbing (up and/or down) and often other similar activities such as lying prone. On occasions it also included cycling. In general the overall description “daily activities” was applied to all cases where either the authors themselves used this description or the set of activities within the work was substantially included within this description. Where the activities included only one or two specific activities from within the “daily activity” group then the specific activities themselves were noted.

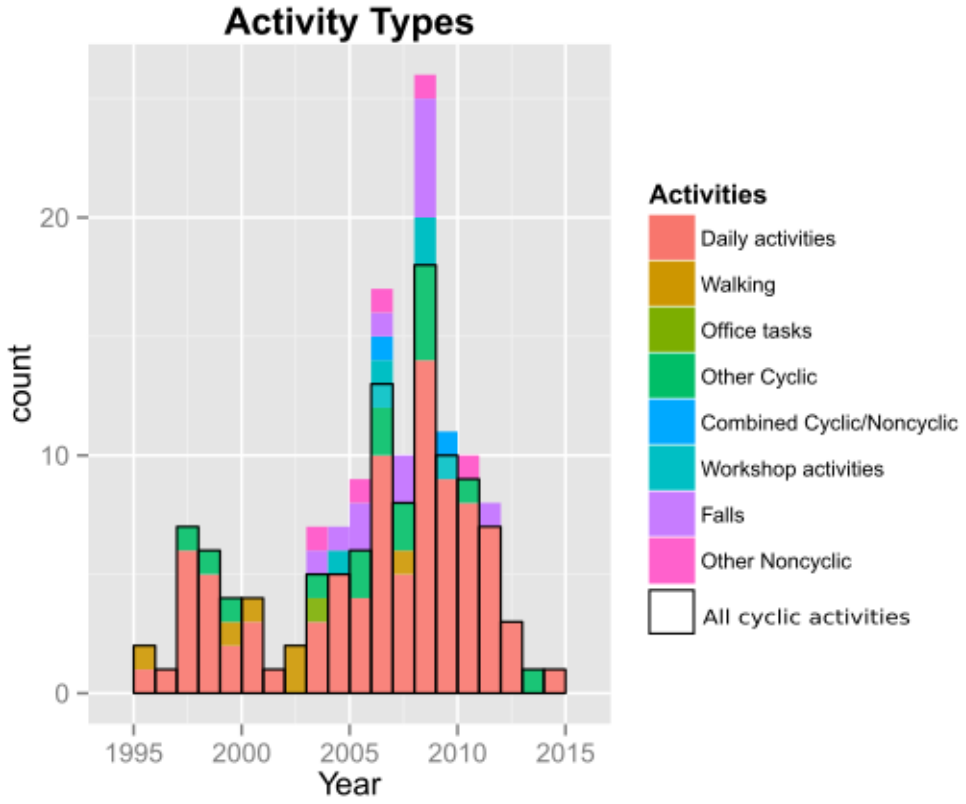


Figure 3.1: Reviewed Activity Types

Interestingly, the next most common activity to be studied was “Falls”, a Punctual activity with 14 of the 137 (10.21%) publications dealing with Falls. In all cases, the researchers who set out to classify Falls were from the Health domain. Three of the 14 Fall publications reported that the inertial data used for classification was not segmented and so classification was attempted using inherent characteristics of the raw inertial data, generally trigger levels associated with acceleration.

Other commonly researched activities are “Walking” with six publications and those that the author classed as “Workshop activities”, also with six publications. Some other, somewhat rarer, activity descriptions included snowboard air time, tremor,

air writing, skiing turns, Bradykinesia (a slowness in execution of movements), delirium motoric sub-types, military daily activities (a super set of daily activities), eating activities and food preparation. Clearly, activity classification research has only attempted to classify a very small number of the activities that would be expected to be encounter in all areas of life.

As can be seen from Figure 3.1, the popularity of reported studies into Daily activities has been there from the start and has generally increased over the middle of the review period, peaking at around 2008 and then seems to drop off somewhat. The author thinks that the rise reflects the increasing interest in inertial sensor based activity classification as classification techniques were developed and refined and that the subsequent fall simply reflects that the material for the review comes primarily from review publications prior to 2013 and so these studies necessarily look backwards and therefore only include works prior to this. While the author's own searches have included the period between 2013 and early 2015 these searches will have been hindered by the time it takes for newer work to be indexed within the various search engines and so the recent drop off may well be less dramatic than shown.

3.2.9 *Scripted versus Unscripted Activity Classification*

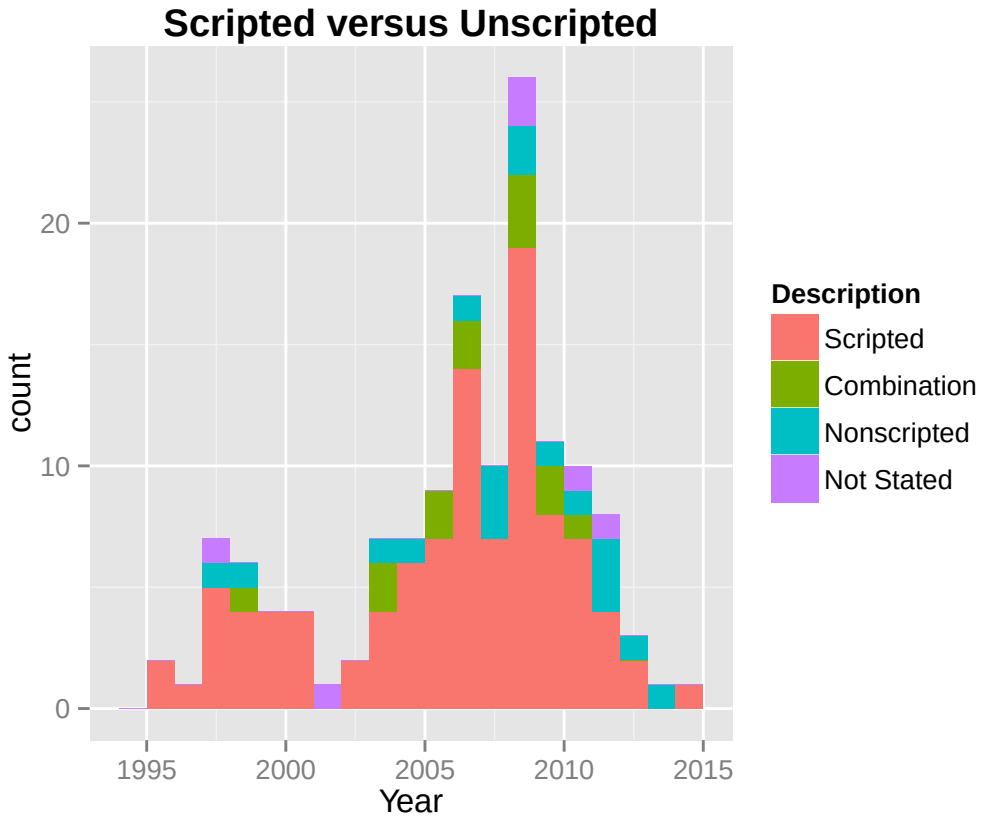


Figure 3.2: Contrasting scripted and non-scripted activities

101 (73.72%) of the publications in this review used scripted activities when collecting data. 13 (9.49%) used both scripted and unscripted activities, while only 17 (12.41%) used only unscripted activities while collecting data. Six publications did not report if their activities were scripted or unscripted. From this the author concludes that the majority of research within activity classification is done using data that is much less representative of real world activities and so may not be easily translatable from research into real life.

Figure 3.2 shows that early work included in this review had a clear preference for data collected from scripted activities and that work based on scripted activities has remained popular. However, from the mid 2000's a number of researchers have used data from unscripted activities and in 2011 almost half of the included works are based on non-scripted activities. The author thinks that this reflects a general understanding amongst activity classification researchers that unscripted activities tend to transfer better to real life situations and, perhaps even more importantly, that data from scripted activities may well contain signal artefacts that reflect the script and as a result, these artefacts are then less likely to appear in real world data.

3.2.10 *Laboratory Based versus Realistically Situated Classification*

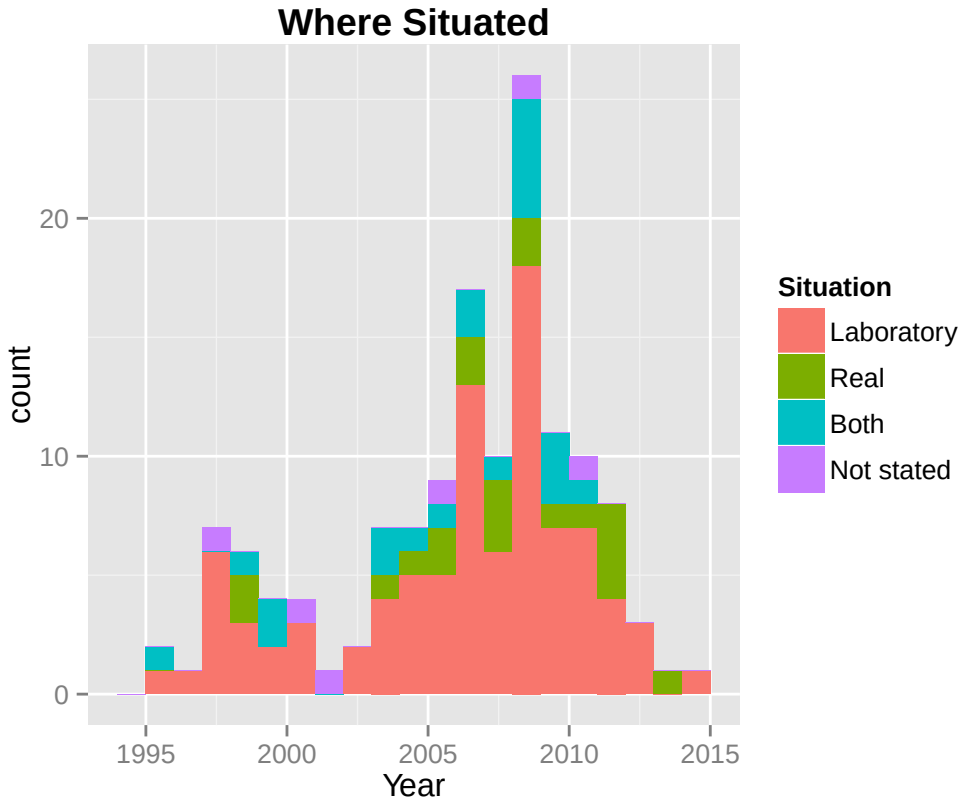


Figure 3.3: Activity Data Collection Situations

Where the data collection is done also has an affect on the ability to successfully translate results from research to real life situations. In general, data collected from real world situations is much more likely to be directly translatable than data collected from laboratory situations. Within this review, 91 (66.42%) of the publications used data from laboratory situations. However, 20 (14.6%) publications reported on research that collected data within a real or highly realistic situation and another 20 (14.6%) reported on work where the data was collected from both laboratory and realistic situations. More recently, researchers have tended to prefer realistic situations for their data collection. Six (4.38%) publications within this review did not state in what sort of situation they collected their data.

While laboratory situated data collection has been popular and easier to do than real world data capture, researchers such as (Foerster et al., 1999), (Parkka et al., 2006) and (Lara & Labrador, 2013) have recognised that data captured from realistic situations was more transferable to real life. The majority of the earlier work utilises laboratory based data capture while real world data capture situations feature more strongly since the mid 2000's.

3.2.11 *Segmented versus Unsegmented data features*

A large majority of the reviewed publications (127 out of the 137, 92.7%) used segmentation techniques to segment the data and then calculated derivative features from the data segments that were then used to classify activities. This is illustrated in Figure 3.4.

Four of the publications did not state if they had segmented their data or not and this could not be worked out from the reported material. Five publications reported using solely unsegmented data and one publication reportedly used both segmented and unsegmented data for classification, together representing six or a mere 4.38% of the reviewed work. This clearly shows the preference for activity classification based on features calculated from segmented data within the current research community.

Of the six works that did not segment their data, three (Bourke et al., 2007, 2008; Nyan et al., 2008) reported using simplistic trigger levels of acceleration to classify the Punctual activity, falls. S. H. Lee et al. (2003) reported using unsegmented data to firstly distinguish between static and dynamic poses and then for static poses used trigger levels on unsegmented data to further classify them into particular static poses. Allen, Ambikairajah, Lovell, and Celler (2006) reported using unsegmented data to classify three static Durative activities (sitting, standing and lying), four Punctual activities (sit-to-stand, stand-to-sit, lie-to-stand and stand-to-lie) and one dynamic Durative activity (walking). The researchers report that they first used a low pass filter to separate approximate gravity components of acceleration from movement components of acceleration and used a number of features (including the gravity and movement values) derived from the unsegmented data to classify the static activities. They also report on using segmented data and derivative features to classify the dynamic activities.

Of most interest to this work, Zappi et al. (2008) report on a technique that converts unsegmented data into one of three values based on trigger levels and which then uses

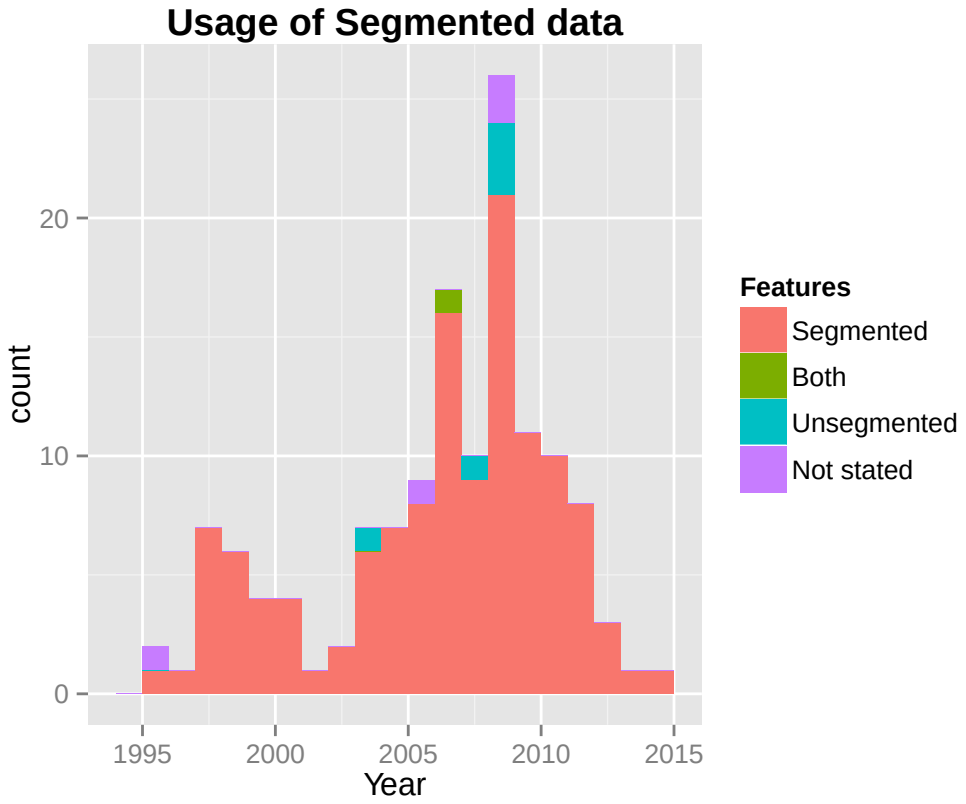


Figure 3.4: Usage of Features from Segmented Data

a series of HMM and a naive Bayes classifier to fuse the output from the HMM and to then classify Punctual workshop activities. The author believes that the approach of Zappi et al. (2008) has merit and in this work the author has started with the idea of using unsegmented data rather than features derived from segments of that data in this activity classification.

3.2.12 *Durative versus Punctual Activities*

Most of the research included in this review deals with Durative activities, 105 of the 137 publications (76.64%) deal only with Durative activities while 19 (13.87%) deal with both Durative and at least one Punctual activity. Eight of the 19 publications that deal with both Durative and Punctual activities have falls as their Punctual activity of interest. 13 publications, (9.49%), deal only with Punctual activities and of these, eight publications also have falls as their activity of interest.

While Figure 3.5 clearly demonstrates the research bias towards Durative activity classification it does also highlight that some research is also being done into Punctual activity classification and that this interest in Punctual activities has developed recently.

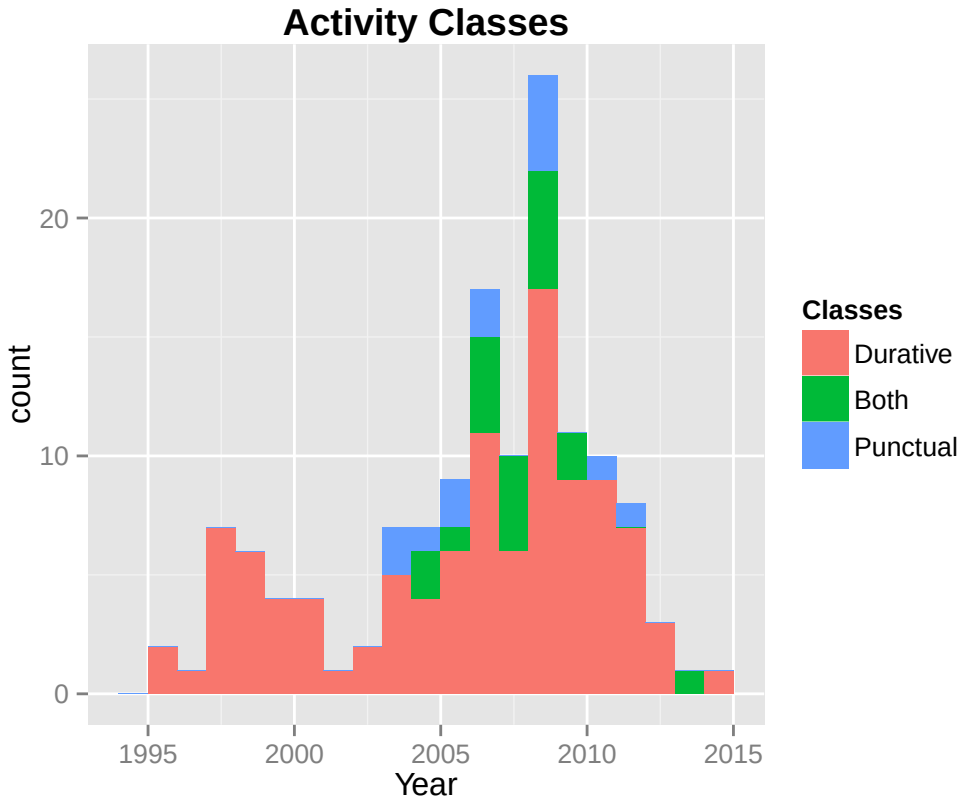


Figure 3.5: Reviewed Activity Classes

3.2.13 *Segmentation of data for Durative and Punctual Activities*

Looking across activity class and segmentation it can be seen that there are 104 publications that deal solely with Durative activities. Of these, 99 (94.29%) publications report segmenting their data prior to classification; four (3.81%) do not state if they segment their data or not and this can not be implied from the description of their work. This leaves a mere two (1.90%) publications that use unsegmented data in some form to help classify Durative activities.

S. H. Lee et al. (2003) describes a process where static poses are distinguished from dynamic activities by "*comparing the amplitude of extracted AC signal from the measured data*". Unfortunately the authors do not describe how they calculate the "*AC*" component of the signal and this process may involve segmentation of the data. However, as this is unstated, it has been assumed for this analysis that their "*AC*" component was calculated without segmentation. Once the "*AC*" component has been subtracted from their data they then use simple amplitude triggers to differentiate the various static poses that hey are interested in. Again, the authors are not clear about how long their "*DC*" signal (original signal - AC signal) must be relatively static for before they deem it static and so it is possible that they are segmenting their data in some form in order to differentiate static from dynamic.

Allen et al. (2006) state that their goal is to "*improve the flexibility and generality of the monitoring system, making it better able to detect and identify* ***short-duration movements*** *and more adaptable to a specific person or device*" and so while the work that they describe in this publication is focussed on Durative activities they are also interested in Punctual activities, particularly falls. Given this goal and their interest in falls, it is perhaps significant or at least interesting that they choose to deviate from the vast majority of Durative activity classification research by not segmenting their data.

Allen et al. (2006) state that their use of time domain features was driven by their desire to capture very short duration activities. The authors separated the accelerometer signal into two parts, body acceleration and gravity acceleration, using a low pass filter with an 0.25Hz cut off. All axis of both parts were then used as part of the input into the classifier, without segmentation. However, the authors also included segmented data as part of their final input. This included slope vectors calculated from an 0.4 second window centred on the current sample.

From this it is concluded that researchers looking to classify Durative human activities almost always choose to segment their input data and calculate features from those segments prior to classification. Although Allen et al. (2006) have shown that where the activity of interest has a very short temporal time span then using either the raw inertial data or simple, unsegmented derivatives from it can provide a classifier that is more responsive to and therefore better able to classify short activities.

Once Punctual activity classification, especially classifiers that classify activities that are very short in duration are considered then this tendency to use unsegmented data becomes slightly more popular. In this review 19 publications reported that they set out to classify both Durative and at least one Punctual activity. Three of these 19 (15.79%) studies used unsegmented data and two (40.0%) of the five studies that had falls as their Punctual activity of interest used unsegmented data. Another 13 publications from this review set out to solely classify Punctual activities. Of these, one (7.69%), used unsegmented data as input into the chosen classifier. Looking across all 32 publications that reported classifying Punctual activities, four (12.5%) used unsegmented data and three (23.08%) of the 13 studies that set out to classify falls used unsegmented data as input.

Most of the works from this review that utilise unsegmented data for classification and all of the works that seek to use unsegmented data to classify falls use a simple heuristic based on trigger levels as the basis of their classification. However, one work, Zappi et al. (2008) uses a more sophisticated technique and is of interest. The authors of this work have used two trigger levels to transform accelerometer readings into one of three states. These three states are then input into discrete HMM so as to make use of the temporal sequence of the time series data.

3.2.14 *Review conclusions*

This review is not complete, the number of works within this review that look at classifying Punctual activities is small and the number that use unsegmented data is even smaller, however, there seems to be enough usage and logic to using unsegmented

data for Punctual and especially short Punctual activities to suggest that the use of unsegmented data as input for classification should at least be considered when attempting to classify Punctual activities. None of the reviewed works has attempted to classify human activity using a RNN as a tool that takes advantage of the spatio-temporal nature of the data. This work would then appear to be novel in this regard.

3.2.15 *Why is Activity Classification useful?*

Section 2.3 outlines most of the reasons why ACS are useful, summarising that, a knowledge of a persons current activity provides context to any computerised system that may be dependent on that activity in some form. From a Human Computer Interface (HCI) point of view as computer systems become more personal and particularly when they are wearable systems then a knowledge of activity allows that system to be "smarter" and possibly easier to use. Potentially allowing such a wearable system to automatically adapt its behaviour to suit the current activity. This is consistent with Weiser (1993) in his seminal work.

Activity knowledge is also useful in health monitoring and this is supported by the earliest interest in inertial based ACS by health researcher pioneers. More recently, with improvements in both hardware and software, industrial and commercial applications with an audit focus have gained currency, initially as a means of monitoring compliance during automobile assembly. Alongside this has been a growing interest by individuals in quantifying their own activities for fitness purposes as demonstrated by an increasing number of wearable activity monitors such as those from Fitbit ®. Lastly and importantly, a knowledge of activity is essential for computerised coaching systems whether in sport or other areas.

3.2.16 *Activity Classification is challenging*

While Human activity classification is useful it is also a very challenging area of research. S. J. Preece et al. (2009, p. 3), states:

> The automated identification of activities using body-worn sensor data is a challenging area of work. Apart from the obvious practical limitations on the number, location and nature of sensors that people will tolerate, there are several issues that directly impact the success of any given algorithm.
>
> Problems arise due to the variability in sensor characteristics for the same activity across different subjects and for the same individual. Errors can also arise due to variability in sensor signals caused by differences in sensor positioning and from environmental factors such as sensor temperature sensitivity. Any successful algorithm must overcome all these factors.

From this it can be concluded that Activity Classification is both useful and challenging and therefore a suitable area of research for a thesis such as this but there is

already a reasonably large body of work that addresses this area and this work addresses an area within Activity Classification that represents a gap in current knowledge. From Section 3.2.6 it can be seen that within the field of activity classification most researchers have chosen to work with Durative activities. Until recently, short, Punctual activities have not been researched beyond the area of Falls.

In addition, what research that has been done on Punctual activities has either used simplistic heuristic techniques in most research on Falls or has tried to use techniques that were developed to classify Durative activities. Section 3.2.6 highlights two works that had attempted sophisticated ways of classifying Punctual activities. This works sits alongside those earlier works and adds to the knowledge in this area. One of the reasons that the author thinks that there has been little prior work done to classify Punctual activities is that Punctual activities are probably even harder to classify than Durative activities.

3.2.17 *Why is Punctual Activity Classification hard?*

If it is accepted that human activity classification is a difficult and active area of research then Punctual human activity classification is even more difficult and is a largely avoided area or research for the following reasons.

Rarity

Many Punctual activities are very rare. For example, getting on a horse usually takes 3–5 seconds and often only happens once during a riding session that may last for anything from [typically] 40 to 120 minutes. Looking at the extremes in this example would have a classifier searching for an activity that is 0.027% of the total data.

There is general agreement that finding these (often rare) events in continuous data is a challenge. By definition, Punctual activities are of short duration and so spectral quantization techniques used successfully with segmented data and Durative activities are not reliable (J. A. Ward et al., 2006). In particular sliding window based techniques can easily miss important data unless supplemented with additional constraints and additional data such as the studies from the successful *Production Line* studies out of ETH's Wearable Computing Lab in Switzerland and similar studies elsewhere.

Finding these relatively rare sequences amongst all the irrelevant data can be difficult. "*... reliable recognition is still an open problem. The main difficulty lies in the fact that large segments of (random) non-relevant activities often occur between activities meaningful to the task*" (J. Ward et al., 2006, p. 6).

This is reiterated by Junker et al., (2008, p. 2) when they state "*For example, in between the actual* (Punctual) *activity a user might fetch tools, drink, chat with another person or just scratch the head. As a consequence, the task at hand can be described as activity spotting. It is widely recognised as a particularly complex domain of activity recognition and is still an open problem*".

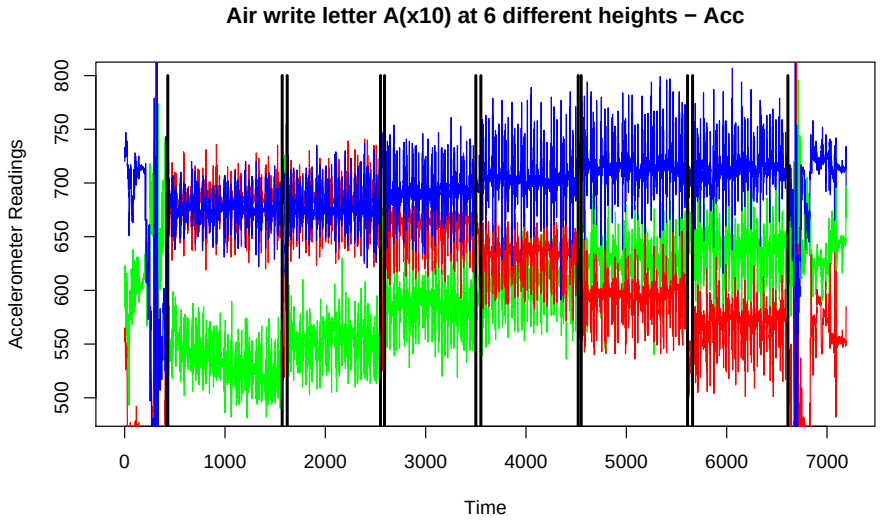
Junker et al. (2008, p. 1) also highlight that this area of Punctual activity classification is difficult across both vision-based and inertial-based systems when they state "*One area where little progress has been made so far, is the spotting of sporadically*

occurring activities in a continuous data stream. This is known to be difficult, even if complete trajectory information is available from a vision system. It is even more difficult in a wearable sensor-based environment".

Techniques that rely on training a model by minimising a cost function, such as this work, are faced with issues of how to effectively minimise a function across a very large base class that may involve 99–99.97% or more of all data points while still distinguishing the very rare activities of interest. It is observed that Punctual activities occurring in real–life situations are sometimes extremely rare and within the real–life data used in this research, this is most assuredly the case. This rarity clearly contrasts Punctual activities with Durative activities as by definition, Durative activities are longer by nature and are therefore expected to be more common.

Recognising that rarity creates classification issues some researchers deal with these issues by not attempting to classify Punctual activities. In other cases, researchers deal with rarity by scripting Punctual activities in a laboratory or simulated real–world environment so that they can control the amount of non-Punctual activity and thereby making the Punctual activities more common with their data. In yet other cases, researchers manually pre-select data with Punctual activity within it and exclude the remaining data to allow for effective classification.

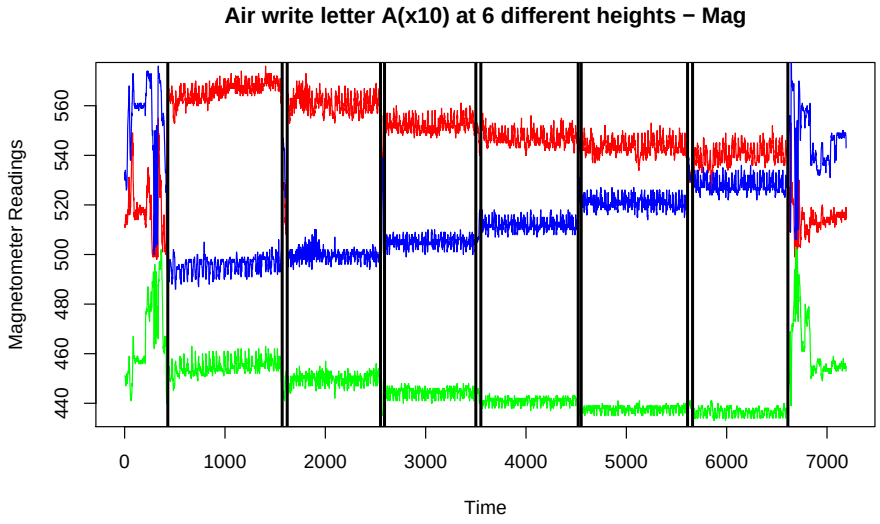
Sensitivity to orientation



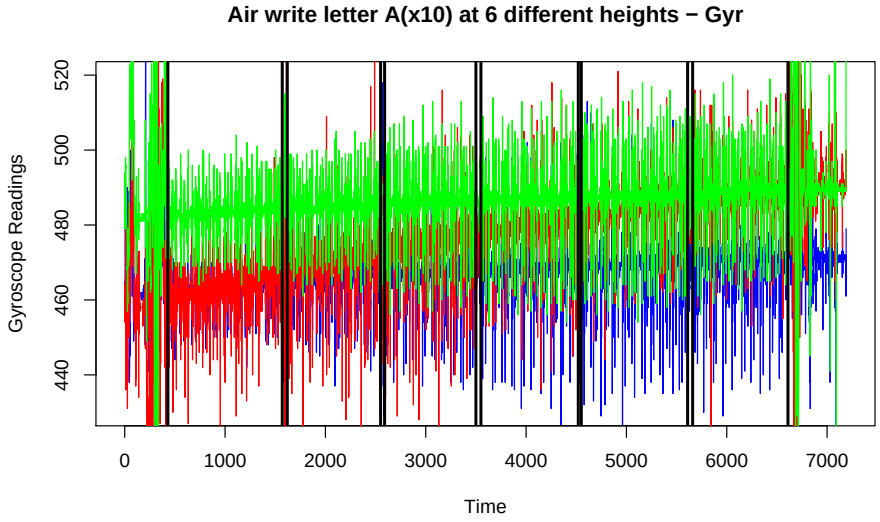
(a) Accelerometer output

Accelerometer and Magnetometer data are highly sensitive to orientation and Accelerometer data is sensitive to the gravity vector (see Figures 3.6a, b & c). Gyroscope data is slightly sensitive to orientation and there is no gravity vector to complicate things. Differences in inter-individual dimensions (e.g. participant height) and technique acerbate orientation changes.

Figure 3.6(a) plots the output from an accelerometer in an experiment where a repetition of ten capital A's were written on a whiteboard at six different heights.



(b) Magnetometer output



(c) Gyroscope output

Figure 3.6: Air–writing A's at six different heights

On the figure the vertical black lines separate each set of repetitions. It is obvious from this figure that the X and Y axes (red and green) change markedly as the height at which each repetition of A's is drawn changes but it can also be seen that even the Z axis (blue) changes as the height change. As a result, raw accelerometer data can be difficult to classify reliably without some means of correction for orientation changes.

Figure 3.6(b), page 50, plots the output from a magnetometer for the same experiment. Here it is equally obvious that all three axes change, in some cases markedly, as the orientation of the writer's hand changes with changes in height. Again, it would be difficult to use the raw magnetometer data without some means of accounting for the orientation changes.

Lastly, Figure 3.6(c), page 50, plots the output from a gyroscope for the same experiment. In this figure there is no obvious stepped change in the three axes as the writer's hand changes in height and orientation but it is possible to see a marked drift in one of the axes (red) and a slight drift in the other two (green and blue) over the time of this experiment. This drift needs to be accounted for if gyroscope data is to be used reliably for classification in close to raw format.

This sensitivity to orientation is one reason why motif type classification using raw inertial data is seldom attempted. However, in the case of this work, the author has previously noted from (Hunt, 2009) that the standardisation of of mounts and dismounts across the equestrian disciplines results in an almost standard orientation for the rider's right hand during at least part of the mount and dismount activities. This standardisation gives a reason to think that using close to raw data may lead to successful classification for the mount and/or dismount activities.

3.2.18 *Reservoir Computing techniques*

RC techniques are a form of RNN. There are two main classes of RNN's according to Lukoševičius and Jaeger (2009). One rooted in statistical physics (these include Hopfield networks, Boltzmann machines and Deep Belief networks) and the other, which may be collectively called RC techniques, is rooted in non-linear dynamical systems (these include LSM and ESN). This work is only interested in the properties of RC systems.

While RC techniques have not previously been used as a classification engine for activities they have been used very successfully in closely related, real-world spatio-temporal problems areas such as isolated word recognition (Verstraeten, Schrauwen, & Stroobandt, 2005), speech recognition (Jaeger, Lukosevicius, Popovici, & Siewert, 2007; Verstraeten, Schrauwen, & Stroobandt, 2006), financial forecasting (Ilies et al., 2007), spoken phoneme recognition (Goodman & Ventura, 2006) and the prediction of chaotic dynamics (Jaeger & Haas, 2004). In addition, Verstraeten, Schrauwen, D'Haene, and Stroobandt (2007) put a strong case for RNN's as powerful tools for solving temporal machine learning tasks. Goodman and Ventura (2006) also argue effectively that RNN's are an excellent tool for pattern recognition on multi-dimensional time-series. RNNs incorporate a fading memory and so are able to in-

clude recent time within their model, this enables these models to effectively classify signals with a temporal aspect.

RC is a subset of RNNs that overcome the complex and tedious training issues with slow convergence rates inherent in earlier RNNs by keeping the internal Input and Reservoir neuronal connection weights static and only training the linear output weights (Verstraeten et al., 2007). LSMs and ESNs are two of three generally recognised RC techniques and are both used within this work. The third RC technique, Backpropagation Decorrelation (BD), which was first introduced by Steil (2004) and Schiller and Steil (2005) is not used within this research.

A RC system is composed of two main parts, the reservoir or liquid (a dynamical system that is used to apply the non-linear temporal transformation) and a trainable, memoryless readout function. The dynamical system is composed of a RNN and its purpose is to transform (encode) the spatio-temporal input into a non-linear (high dimensional) spatial form (Jaeger, 2001; Lukoševičius & Jaeger, 2009; Maass, 2010; Maass, Natschläger, & Markram, 2002). The memoryless readout snapshot then includes both current and past events simultaneously in a simple linear form. Figure 3.7 on page 53 provides an analogy of the reservoir used within RC systems and provides a strong hint to the source of the "reservoir" name. In this simple two dimensional snapshot photo an observer can see the current position, direction and recent path of two boats, with the most recent paths being more easily discernible than the less recent, which are starting to fade out. Other patterns are also discernible, possibly additional paths of boats that are now out of frame. the interference patterns within this snapshot demonstrate the "fading memory" effect of such a system as it shows both the current situation and recent past.

Both forms of RC used within this research require a number of parameters to be set that define the characteristics of the reservoir, including such characteristics as size, neural model, synaptic model, spectral radius, leak rate and others. These characteristics are a key component of the model's ability to successfully classify the model inputs. While some work, such as Jaeger (2005); Maass et al. (2002); Verstraeten et al. (2005), has been done on reasonable ranges for such parameters, there is no simple, conclusive set of parameters that are recommended for the use of these techniques. As a result, the successful use of these techniques requires some sort of search across the possible parameter space looking for candidate sets of parameters that work well with both the inputs and the classes that need to be successfully classified. However, the search space and error landscape of the RC parameter set is very complicated often with many local minima (Lukoševičius, 2012) and so searches across the parameter space often require sophisticated search mechanisms in order to reach good enough classification results.

LSMs were first introduced by Maass et al. (2002) and are somewhat more biologically inspired than ESNs and often use "*integrate-and-fire* (artificial) *neurons and dynamically adapting synapses*" (Fette & Eggert, 2005). A LSM has two major components, a reservoir or "liquid" in the form of a Recurrent Spiking Neural Network (RSNN) (Gerstner & Kistler, 2002) and a trainable readout function. A LSM may have multiple, independent, task specific trainable readout functions and so is capable of parallel computation in real-time (Maass et al., 2002). LSM are a some-

Figure 3.7: Analogy of Reservoir Computing using a real reservoir

what more biologically realistic form of RC model, especially when used with Leaky Integrate-and-Fire (LIF) neurons and Short-Term Plasticity (STP) as described by Markram, Wang, and Tsodyks (1998).

In a LSM model the liquid is stimulated by time-varying input signals passed through input neurons, causing (neural) activity in the RSNN that is further propagated through the network due to the network's recurrent topology. As a result, a snapshot of the (neural) activity in the reservoir contains information about the current and past inputs to the system. The function of the liquid is to accumulate the temporal and non-temporal information of all input signals into a single high-dimensional intermediate state in order to enhance the separability between the LSM inputs. The readout function is then trained to transform this intermediate state into a desired system output.

ESNs, on the other hand were first introduced by Jaeger (2001) and utilise a "*discrete time, non-linear, recurrent network*" (Fette & Eggert, 2005) often composed of sigmoidal artificial neurons.

ESN have been employed for a wide range of spatio-temporal real-world problems such as speech recognition Jaeger et al. (2007); Verstraeten et al. (2006), financial forecasting Ilies et al. (2007) and the prediction of chaotic dynamics Jaeger and Haas (2004). However they have not previously been used to classify human activities based on inertial sensor inputs.

According to (Lukoševičius, 2012), the underlying technique for using an ESN is as follows, where, essentially, the ESN is trained to map the inputs onto a target (class) signal: firstly generate a large, random reservoir then run the training data through that reservoir and collect the reservoir activation states. Then compute the linear readout weights from the reservoir using some form of linear regression while minimising the error of the reservoir output versus the target training signal. Lastly classify new instances by using the same reservoir and previously trained output weights to compute a new output signal based on new inputs.

For this underlying technique to work consistently, the reservoir must not amplify the input signal to such an extent that that the output signal becomes chaotic. Instead, the reservoir must possess the *echo state property* (Lukoševičius, 2012). this property ensures that the effect of the input signal will fade over time. This *echo state property* is usually ensured by scaling the the previously randomised recurrent connection weights to an appropriate level (*spectral radius*). It is also normal to scale the inputs by some factor in order to change the degree of linearity or non-linearity of the $tanh$ function and to adjust the neuronal leaking rate to change the amount of *memory* exhibited by the reservoir in order to get better classification results.

Somewhat more formally, the ESN is a neural network consisting of N_u input neurons, N_x reservoir (hidden) neurons and N_y output neurons. The reservoir neurons are either fully or sparsely interconnected with connection weights specified by a weight matrix $\mathbf{W} \in \mathbb{R}^{N_x \times N_x}$. Matrix $\mathbf{W}$ is initialized with random (uniform) weights and then scaled by the spectral radius $\rho(\mathbf{W})$ which is the largest eigen value of $\mathbf{W}$. Generally, all N_u input neurons are connected to all reservoir neurons via connection weights defined by a separate input matrix $\mathbf{W}_{\text{in}} \in \mathbb{R}^{N_x \times N_u}$. The weights of the input matrix are initialized as either -1 or 1 and then scaled by a scaling factor.

At time step t, the input $\mathbf{u}(t)$ is fed into a neural network. The output of all reservoir neurons in the network is computed:

$$\mathbf{x}'(t) = f(\mathbf{W}_{\text{in}}\mathbf{u}(t) + \mathbf{W}\mathbf{x}(t-1)) \tag{3.1}$$

$$\mathbf{x}(t) = (1-a)\mathbf{x}(t-1) + a\mathbf{x}'(t), \tag{3.2}$$

function f being a neuron activation function usually defined as the (element-wise) hyperbolic tangent $\tanh(\cdot)$, and factor $a \in \mathbb{R}$ being a leaking rate that controls the contribution of the previous neural output to its current state.

ESN are an elegant and efficient solution for imposing a desired input-output behaviour onto the network. Input and output vectors at time step t are concatenated and then linearly transformed into the final output of the network:

$$\mathbf{y}(t) = f_{\text{out}}\left(\mathbf{W}_{\text{out}}[\mathbf{u}(t)|\mathbf{x}(t)]\right) \tag{3.3}$$

where $\mathbf{W}_{\text{out}} \in \mathbb{R}^{N_y \times (N_u + N_x)}$ is a weight matrix connecting all reservoir and all input neurons with N_y output neurons, $\cdot|\cdot$ represents the vertical concatenation of vectors and f_{out} is the activation function of the output neurons which is usually chosen as the identity. The learning task is defined as an optimization problem in which the difference between $\mathbf{y}(t)$ and $\mathbf{y}_{\text{target}}(t))$ is minimized. Arguably the most popular

technique of computing matrix $\mathbf{W}_{\text{out}}$ is linear regression or its regularized extension called ridge regression:

$$\mathbf{W}_{\text{out}} = \mathbf{Y}_{\text{target}}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \alpha^2\mathbf{I})^{-1} \tag{3.4}$$

where $\mathbf{I} \in \mathbb{R}^{N_x}$ is the identity matrix and $\alpha \in \mathbb{R}$ is a regularization factor that has to be carefully tuned for optimal results. Matrix $\mathbf{Y}_{\text{target}} \in \mathbb{R}^{N_y \times T}$ contains all vectors $\mathbf{y}_{\text{target}}$ concatenated into a matrix. We note that the connection weights $\mathbf{W}$ are not modified by the learning rule and only the output weights $\mathbf{W}_{\text{out}}$ are updated during training.

The role of the reservoir is the transformation of the input signal into a high-dimensional intermediate feature space represented by the neural network output at any time t. Although linear techniques are then used to transform the feature vector into a desired target output, the mapping of the input $\mathbf{u}(t)$ to output $\mathbf{y}(t)$ is of non-linear nature.

By choosing a straightforward linear learning rule, the training process becomes highly efficient. ESN allow the exploitation of the interesting characteristics of RNN without the need of mathematically and computationally complex training algorithms.

3.2.19 *Advantages of Reservoir Computing Systems*

The fading memory aspect of RC systems (as discussed in Section 3.2.18, starting on page 51) is particularly useful when using RC techniques to classify Punctual activities, particularly Punctual activities that have a wide range of completion times such as the activities of interest within this work, Mounting and Dismounting. When RC classification techniques are paired with unsegmented input (sensor) data and where the reservoirs are large enough, then the issue of how wide/narrow the input data window segments should be becomes mute. While a classifier that uses fixed input data window widths must necessarily make a choice about how wide the window should be and thereby risk choosing a width that is suboptimal for some Punctual activities when they vary in completion time; a RC based classifier need only choose a reservoir large enough to encompass at least the longest possible completion time and in many cases the reservoir will be several orders larger than the minimum needed for other reasons. Of course, there are other techniques that can be used to overcome the issue of matching input data window sizes to varying activity completion times and so RC classification techniques are not unique in this area, however they do represent a useful and novel approach to this problem.

RC systems project their input into a high dimensional space and then use a simple, memoryless readout snapshot that includes both current and past events simultaneously. This snapshot effectively forms a hyperplane between classes. This means that once the system is trained it can classify with very fast response times. This is very useful in systems such as the proposed wearable coaching system because a faster classification system enables the "coach" to respond faster to activity changes.

Also, several different readout functions can each be trained to perform different classification tasks on the same input and reservoir so that the reservoir need only be calculated once. Each activity class has its own readout and output weights. This allows the parallel classification of more than one activity. In Verstraeten et al. (2007) the researchers concluded that their LSM for spoken digit classification provided comparable performance to an optimised HMM on clean data and decayed slower with additional noise levels. In addition, their LSM could extract additional data from the same reservoir (speaker identification) whereas the HMM could not do this without further computation. This ability to classify reliably in the presence of noticeable noise is especially useful when used with inertial data which tends to be particularly noisy.

In many ways, RC techniques are similar to other kernel-based machine learning techniques, most prominently the Support Vector Machines (SVMs) in that they project their input into a high dimensional space and then generate intermediate feature vectors for classification but RC techniques are inherently temporal whereas SVMs are not. By choosing a straightforward linear learning rule, the training process becomes highly efficient. Maass et al. (2002) showed that a sufficiently large LSM can approximate any time invariant filter with fading memory and that they are ideal for real-time processing of time-variant inputs and real-time classification.

RC models have been shown to be effective at classifying spatio-temporal data in a number of situations and have been shown to dramatically outperform other classification techniques in some instances (Lukoševičius & Jaeger, 2009). The RC model architecture is biologically plausible, seemingly similar to observed characteristics of some parts of mammalian brains, especially speech related centres (Lukoševičius & Jaeger, 2009). RC models have been shown to be particularly effective in situations where the input data is noisy (Lukoševičius & Jaeger, 2009), a fundamental characteristic of inertial data collected from body mounted inertial sensors. As such they represent what seems to be a plausible tool for classifying human activity based on inertial sensor data.

Of the two RC techniques considered within this research, LSM are more similar to the neuronal models observed within mammalian brains than ESN and are generally considered to be more powerful computational models. However, the increased computational power over ESN comes at a cost of additional complexity and, perhaps more importantly within an iterative research environment such as this, generally require much more computation resources when modelled on general purpose digital computers.

3.2.20 *Initial Literature Review Summary*

This research looks at using machine learning techniques to classify Punctual activities. The current literature does not, in general, cover this approach to activity recognition with inertial sensors data very well except for some specific work on the single punctual activity, Falls, and so this is seen as a valid gap to be investigated.

This work has replicated the concept of Punctual and Durative activities as major sub-classes of activities from video based activity classification research and intro-

duced that concept into inertial data based activity classification research where it has not been present.

Punctual activities are seen to have important differences from Durative activities. In particular, Durative activities are cyclic whereas Punctual activities are non-cyclic; Durative tend to be longer while Punctual tend to be shorter and in many cases the temporal characteristics of Punctual activities are often key to their classification whereas for Durative activities often it is their frequency characteristics that are the key to classification. These different characteristics mean that in many cases when a single classifier is used for both types of activity then one type is often not classified as accurately as the other. Other than when classifying falls, punctual activities are less represented within inertial data based activity classification research and specialised techniques designed specifically for non-fall punctual activities are rare within the current literature.

The Punctual activities of interest within this work are mounting and, to a lesser extent, dismounting a horse and are taken from the domain of equestrian sport. This work defines these activities in enough detail to enable consistent classification. The desire to classify Punctual activities and the choice of the particular activities of mounting and dismounting reflect the wider goal of developing components for a possible wearable coaching system that could be applied to equestrian sport and possibly to wider areas that require real-time coaching via wearable systems. Wearable coaching systems are likely to be adaptive systems that use contextual information to assist with adaptation as they are used. It is also likely that in order to provide real-time feedback that wearable coaching systems would need some ability to predict the likely next activity so that as the activity is being done it can be compared with a "model" or good example of that activity so that possible feedback can be given quickly enough to modify techniques as the activities are still being done.

Given the goal of Punctual activity recognition, a number of techniques could be used to isolate and recognise the Punctual activities. Within this research, it is intended to use RC techniques for Punctual activity recognition, this is a novel approach and has not been previously reported on in the literature.

In addition, most other Punctual activity classification systems pre-process the data either using a windowing technique (often with a fixed window size) or use some sort of segmenting scheme to reduce the data volume. This makes it harder (but not impossible) to classify the input data stream in real time. RC techniques do not need to have the input data stream pre-processed in this manner. RC techniques utilise an inherent 'window' based on how long the wave interference pattern resonates but this provides more flexibility than most fixed windowing techniques and is quite efficient. This gives RC techniques a potential advantage and not segmenting the input data is a somewhat novel approach.

Many Punctual activities are naturally of variable length. This temporal variability complicates spectral type quantization techniques and possible motif based classification techniques. Psarrou et al. (2002) state "*The temporal window of an activity cannot be constrained. An activity therefore, needs to be recognised based on accumulated information and non-linear temporal scaling*". As a result this work uses close to raw inertial data and a technique that accumulates the data over time. While

other machine learning techniques such as HMM are able to accumulate information over time the recurrent classifier within RC techniques have desirable characteristics that may be preferred for Punctual activity classification.

Sporting domains and in particular equestrian sport has strict "rules" and "preferred techniques" that reduce the search space for many activities within those domains. This is an advantage when classifying activities in those domains and gives some encouragement that this approach may ultimately be successful. Specifically, this study looks at applying RC techniques to the classification of inertial data signals from simple inertial sensors worn on people in real world situations within the domain of equestrian sport. If possible, the RC techniques will be used to automatically classify mounting and/or dismounting from within equestrian sport. To the best of the author's knowledge no other reported works have sort to classify activities within equestrian sport.

3.2.21 *Implementing a Punctual Classifier*

The proposed Punctual classifier is designed to operate on pre-recorded data, that is, it is not designed to operate on-line. The overall system concept, on the other hand, requires a classifier that operates on-line if real-time feedback is to be produced. Transitioning the (hopefully) working classifier from this work into an on-line component is outside the scope of this research although the assumption is that such a transition is at least possible. Research such as Jaeger (2003); Jaeger and Haas (2004); Soh and Demiris (2014); Venayagamoorthy (2007) demonstrates the use of ESN for on-line and incremental learning and so there can be some confidence that moving the proposed classifier to an on-line implementation is possible.

3.3 POST DEVELOPMENT LITERATURE REVIEW

After the development phases were complete a second, shorter review was made of the relevant inertial data based activity classification literature to see if any of the points raised in the initial review need amending. That shorter review follows.

3.3.1 *Post Development Review methodology*

The methodology followed a similar process but in this case the author did not rely on widely cited reviews to form the base of the literature that was looked at and instead relied on a direct search of the substantial collection of research literature compendiums available at the author's university library, supplemented by a search via Google Scholar. The author believes that while the list of reviewed work is not absolutely complete that it is comprehensive enough to allow valid conclusions to be drawn. A summary of the modified procedure appears below:

1. The use of the author's own searches via directly accessible databases such as IEEE, ACM and the other widely used academic literature compendiums plus Google Scholar to find publications

2. To exclude all the literature that has previously been reviewed, that is, this review only covers work not previously considered
3. to cover the period from 2013 to early 2017, subject to the prior proviso
4. To include only inertial based activity classification publications
5. To exclude gesture recognition and motion tracking work
6. Where the same or very similar authors re-publish the work under differing titles then only one, representative publication, has been included. However, new work from research using the same data has been included where it is deemed different enough. Of course, decisions to include or exclude work is based on interpretations and is open to challenge

The list of reviewed publications includes 52 publications from 2013 to early 2017. The initial review noted a drop off in research and in particular after 2013 and so this review looks back to that period to see if, from this latest search, if that drop off was real or not. The individual works are not necessarily referenced directly within this section but are all listed within the references chapter. Works used within this part of the review are noted with an asterisk in the references chapter. Where specific characteristics of a particular publication are commented on then that work is directly referenced in this section and is shown without an asterisk in the references chapter.

The works considered within this review are characterised by a number of different attributes, these attributes are much the same as those considered during the initial review but also contain three new attributes that were included either because of the frequency with which they are now appearing (Smartphone and Neural Net classifier) and another attribute, Frequency, that notes the frequency at which the data was collected:

Author - The author(s) of the publication that was reviewed

Year - The year that the reviewed publication was published

Activities - A short description of the activities that were researched within the publication

Segmentation – Was the inertial data segmented prior to classification

Domain – The (implied) primary domain or background for the researchers who published the works

Class – Durative, Punctual or both. This was derived from the author's own judgement.

Situation – Laboratory or Real World situation; note that the author interprets "laboratory" in its widest sense

Scripted – A note of if the activities that subjects performed were scripted or completely free-form

Numbers – The number of participants who performed the activities during the data capture part of each piece of research

Smartphone – (New) If a Smartphone was used to collect inertial data

Frequency – (New) The frequency at which the inertial data was collected

Neural Net – (New) If a neural net was used as one of or the only classifier

In addition a general notes field was kept for any particularly noteworthy point within the publication.

3.3.2 *Activity Types*

A plot of the frequency of various activity types encountered during the review is shown in Figure 3.8 on page 60 and attention is drawn to this plot during this part of the discussion.

In the initial review it was noted for Activity type that the single largest common description was "Daily activities", at 64.23% of all work reviewed. In this later review this has increased to 75%. This is probably related to the research done by what I have called the "Behaviourists", who are attempting to improve some part of the classification process using an experimental research technique. These researchers base their work on existing, well known activity types and often use publicly available datasets to allow their work to be compared with other work. This theme will be explored again when we consider the background domain of the researchers on the following pages.

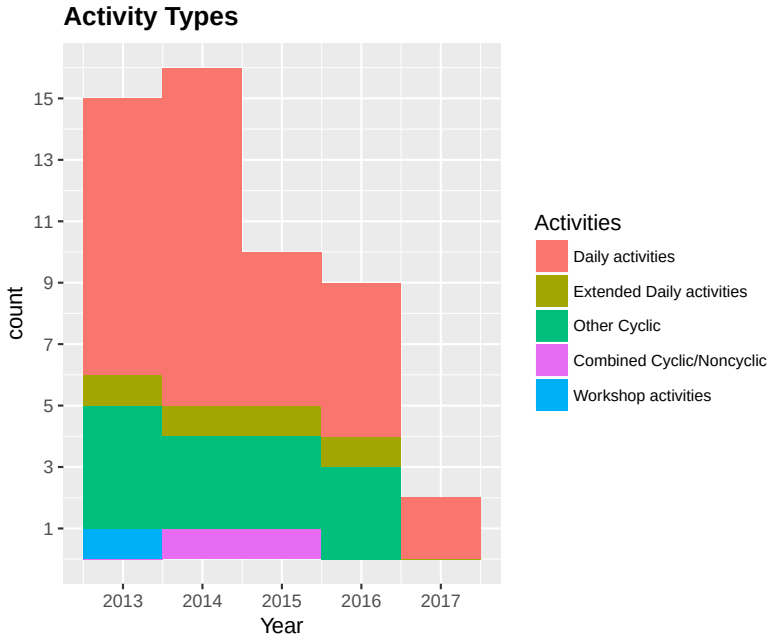


Figure 3.8: Reviewed Activity Types

Notable new and novel activities encountered in this review include Hoettinger, Mally, and Sabo (2016) who attempt to classify wave riding (surfing) and Li, Chen, Chen, Huang, and Chu (2013) who attempt to classify human oral activities such as eating and talking by embedding a sensor in a false tooth. Other interesting activities

within the reviewed works include automobile assembly, (Vollmer, Gross, & Eggert, 2013) and sports activities, (Margarito, Helaoui, Bianchi, Sartor, & Bonomi, 2015).

Falls only appear in two of the publications encountered during this review, one, (Y. Chen & Xue, 2015) used a currently popular neural network based classifier, Deep Learning to classify the falls based on the raw sensor data and had reasonable success. This overall process shows some similarity to the process that was followed in our own work. In the other case involving falls, (Mandal, Happy, Behera, & Routray, 2014) the researchers followed a more traditional approach of segmenting their data, calculating features from the segments but then placed the features into "bins" before classifying them using a SVM. Unfortunately, this work used a script during data capture that imposed an artificial five second pause before each fall and so while this work had reasonable success in classifying the data used it is highly unlikely that the same or a similar classifier would work as successfully in a real world situation where five second pauses before a fall are uncommon.

This more up to date review confirms the authors earlier observation that only a very small minority of all possible human activities are currently researched. This review confirms the earlier caution that the apparent drop off in research in inertial data based human activity classification that was noted in the earlier review was an artefact of both the previous methodology that was used in the prior review and of the slight delay in works being indexed after publication. There is no notable drop off in research in this area.

3.3.3 *Scripted versus Unscripted Activity Classification*

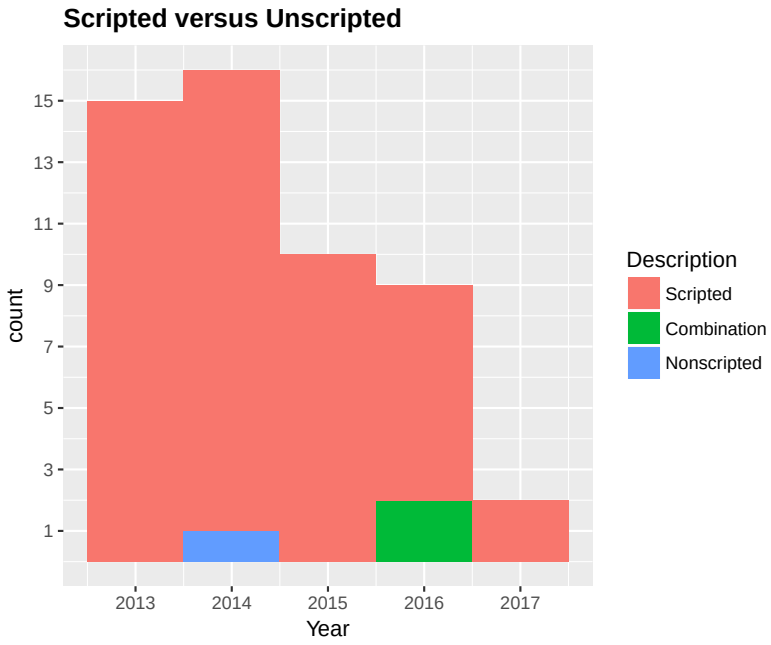


Figure 3.9: Contrasting scripted and non-scripted activities

49 (92.4%) of the publications in this review used scripted activities when collecting data. 2 (3.8%) used both scripted and unscripted activities, while only 1 (1.9%) used only unscripted activities while collecting data. One publications did not report if their activities were scripted or unscripted. This overwhelming bias towards collecting human activity data from scripted situations reflects the large time commitment needed to collect data in an unscripted natural environment and the personal privacy issues associated with constantly monitoring individuals as noted by various authors including Bowen, Hinze, and Cunningham (2015). This current overwhelming bias is shown pictorially in Figure 3.9, page 61. The one area of inertial data based human activity classification that consistently uses unscripted data is that associated with (instrumented) Smart Homes research such as the CASAS Project described by (Cook, Augusto, & Jakkula, 2009). In general, however, most of the published research that has come out of these environments has looked at classifying meta activities such as "eating", or "washing dishes" and as meta activity classification has been excluded from our work, the publications that classify meta activities are not considered within this review except where they obviously include simpler activities or are notable for some other reason.

3.3.4 *Laboratory Based versus Realistically Situated Classification*

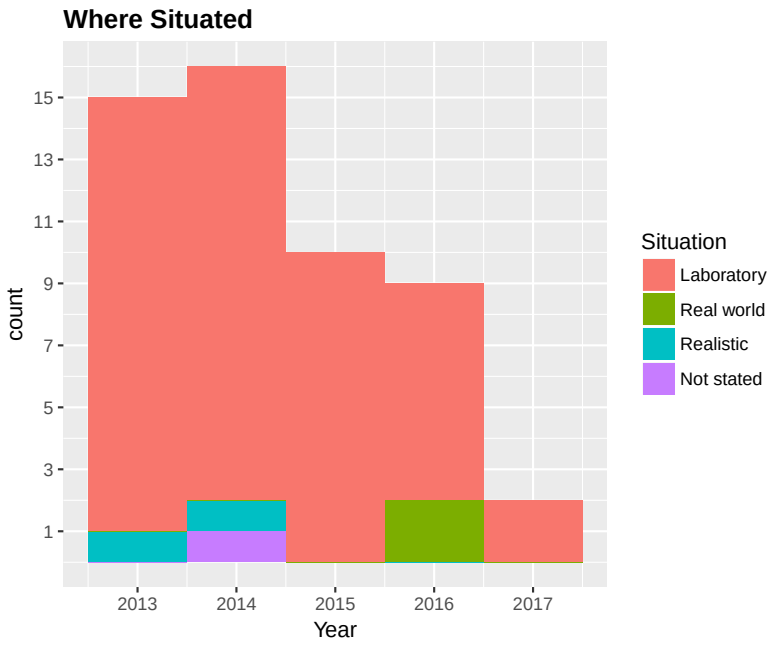


Figure 3.10: Activity Data Collection Situations

Looking at the situation where data was captured we see a change back to laboratory based data capture sites versus real world situations with 47 publications (90.3% of those reviewed) using data captured in a laboratory situation. Four publications captured their data in a real world situation, of those, two situations had subjects follow a proscribed script within their own home. One publication did not state where their data was captured. It is suggested that the change back to an overwhelming

majority of data being captured in laboratory environments is related to the majority of the research being done by "Behaviourists" that are focussed on using experimental techniques to compare their changes to established inertial based human activity classification processes against earlier work. This supports the comments in Section 3.3.2.

3.3.5 *Segmented versus Unsegmented data features*

Once again a large majority of the reviewed publications (50 out of the 52, 96.1%) used some sort of technique to segment the data and then calculated derivative features from the data segments that were then used to classify activities. This is illustrated in Figure 3.11.

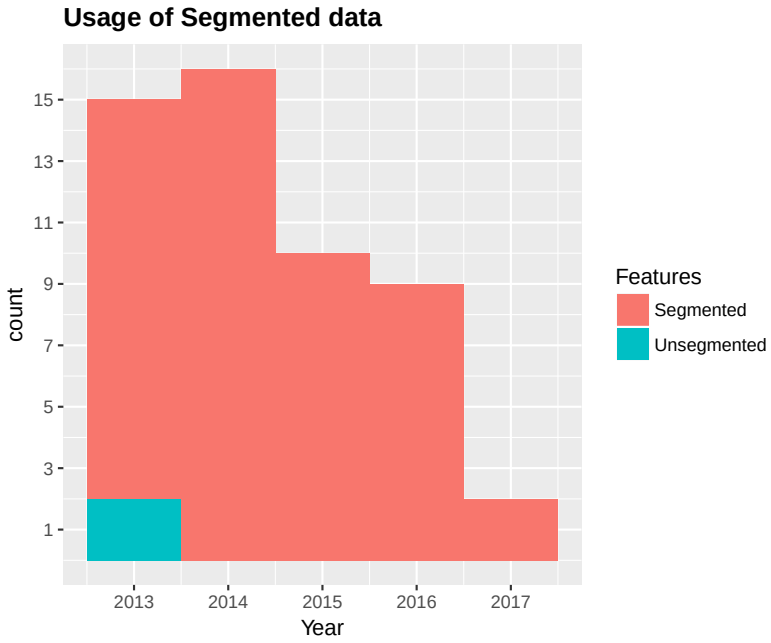


Figure 3.11: Usage of Features from Segmented Data

Two publications reported using solely unsegmented data and one of these (Vollmer et al., 2013) re-used the data collected during Zappi et al. (2008) where it was also used unsegmented. There is a prior comment on this work in Section 3.2.11 on page 43. The other publication that reported using unsegmented data for classification was (Derawi & Bours, 2013) and in this case the authors reported that they used a Dynamic Time Warp (DTW) process to classify activities. Again this clearly demonstrates the bias towards activity classification based on features calculated from segmented data within the current research community.

3.3.6 *Durative versus Punctual Activities*

Once again, within this time frame, 41 out of 52 (78.8%) of publications reported that they classified Durative activities, 10 (19.2%) reported classifying both Durative

and Punctual activities and only one publication (1.9%) reported classifying only Punctual activities. The solitary publication that only classified Punctual activities was Vollmer et al. (2013), the work that also used unsegmented data and which looked at automobile assembly tasks from an audit perspective.

Of the ten publications that considered both Durative and Punctual activities, Noor, Salcic, and Wang (2016) reported using an adaptive sliding window technique to segment their data because they found that using a fixed window size for segmentation caused their classifier to misclassify their Punctual activities. This supports the contention that Punctual activities often need special techniques because of the attributes that differentiate them from Durative activities. One of the other publications (Y. Chen & Xue, 2015) looked at falls as well as more common daily activities and their research is notable for their use of raw, unsegmented data and a Deep Learning based classifier. This, again, has similarities to our own proposed approach and strengthens it. Mandal et al. (2014) also considered falls as well as daily activities but their work is not otherwise noteworthy.

Lubina and Rudzki (2015) reported that their classifier produced significantly reduced accuracy when classifying their Punctual activities, compared with their Durative activities, further supporting the contention that Punctual activities require specialised techniques in order to get higher accuracy levels. Gao, Bourke, and Nelson (2014) mainly looked at the common (Durative) daily activity set but unlike most other researchers they also looked at the transition between some of the Durative activities such as sit–to–stand and these formed their Punctual activities. However this work is spoiled by the manual pre separation of each set of activities into separate files for classification and so their classifier only had to choose between a small subset of known activities and did not need to "find" the Punctual activities from amongst other data. Gupta and Dallas (2014) also considered transitions between daily activities as well as the daily activities themselves but they only worked with two subjects, in a laboratory situation and following scripted activities and so they run the risk that their classifier was using signal artefacts from the scripts during classification. Moufawad el Achkar et al. (2016) also looks at transitions such as sit–to–stand as well as the usual daily activities and in their case they use threshold values to classify their Punctual activities and features calculated from segmented sensor data for their Durative activities, again highlighting that other researchers have concluded that Punctual activities need special consideration. Fida, Bernabucci, Bibbo, Conforto, and Schmid (2015) found that with their mix of durative daily activities together with transitional Punctual activities that they needed to shorten their segmentation window considerably from the optimum for Durative activities in order to accurately classify their Punctual activities.

Interestingly, Palumbo, Gallicchio, Pucci, and Micheli (2016) cites this authors prior work in their research which looks at falls along with daily activities. Unfortunately, they also manually separate out each class of activity into a separate file prior to classification and so their classifier has a simpler task.

While Figure 3.12 clearly demonstrates the research bias towards Durative activity classification it does also highlight that some research is also being done into Punctual activity classification and that this interest in Punctual activities is ongoing.

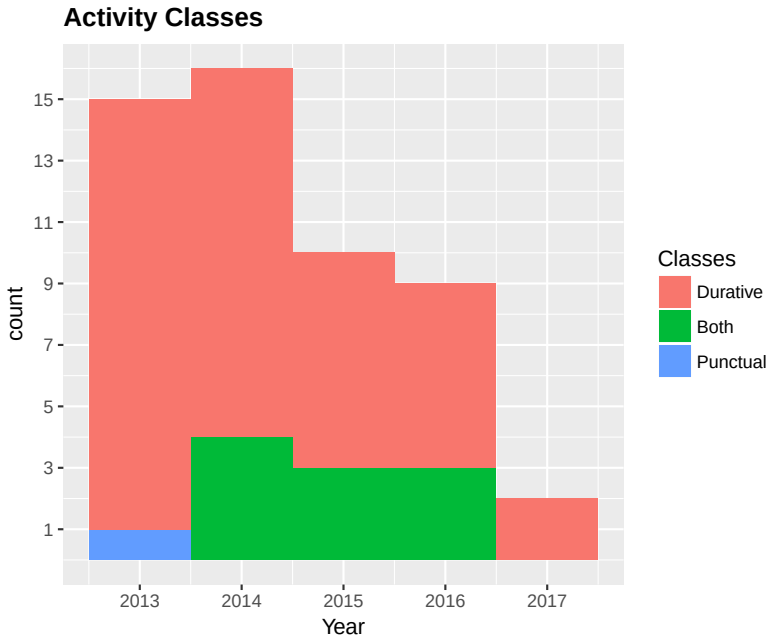


Figure 3.12: Reviewed Activity Classes

3.3.7 *Other things of note from the post development review*

A large number of recent publications (26 out of 52, 50%) have used Smart Phones to collect their data for classification. In a number of cases the only really novel aspect to this research has been the use of Smart Phones and in general they have repeated early research done with stand alone sensors. However there are also some very novel data capture sensors within this group such as Li et al. (2013), tooth sensor and Moufawad el Achkar et al. (2016), shoe sensor. The data capture frequency within the considered publications ranges from 5Hz (Abdallah, Gaber, Srinivasan, & Krishnaswamy, 2015) to 400Hz (Wenlong & Sazonov, 2014) and so, clearly there is little agreement amongst these researchers on what is an ideal frequency for data capture. There seems to be an increasing interest in using neural network based classifiers for inertial data based activity classification with 20 out of the 52 (one not stated) using neural networks. However, we have not done a historical analysis of this and so our comments are anecdotal only.

3.3.8 *Post Development Review conclusions*

This review, conducted after the development of the classifier artefact that has been the focus of this work has confirmed and strengthened the initial literature review and its conclusions. That is, that Punctual activities are recognised by other researchers as being different from common Durative activities and the differences are significant enough to require special techniques aimed at their differences in order to achieve satisfactory classification rates for them. Although Punctual activities are recognised as being different there has not been a clear nomenclature or definitions for them within

inertial data based activity classification research. While there is some work that addresses Punctual activities these activities are poorly researched compared with Durative activities. Where Punctual activities are researched then other researchers are using some of the same techniques proposed within this work such as unsegmented sensor data.

Chapter 4

METHODOLOGY AND APPROACH

This chapter describes the usage of and reasons for choosing Design Science as the methodology for this research. The sensor data used within this research was collected during a prior, Masters project (Hunt, 2009) and so some aspects of data capture are not covered in this chapter but are available from the earlier publication.

4.1 INTRODUCTION

The Background chapter (2) discussed the wider view of a Wearable Coaching System and how that view requires a Classifier system. Further, it was shown that a Classifier system needs to address both Durative and Punctual activity classification. The goal of this research was set to focus on a Punctual Activity Classification system. As a result, the work described in this thesis is focused on creating a successful Punctual Activity Classifier. More specifically, the overall goal of this research is to construct an ACS that successfully classifies the punctual activity *Mounting* within equestrian sport. With this in mind, the outcome of this research will be a, hopefully useful, software artefact. Following from the desire to produce a useful artefact, the taxonomy outlined in Jarvinen (2000) was used to choose Design Science as the appropriate methodology to produce this outcome.

This research consists of three phases. The first phase will be a proof–of–concept phase where RC methods for punctual activity detection from inertial data will be used with both synthetic data and then data from scripted activities. The second phase will entail finding a reliable classifier for the data from the scripted activities performed within the laboratory environment and the building of a reliable tool set of RC code and model parameter optimisation processes that allow for reliable and relatively simple model building. The third and last phase will entail using the tools and knowledge gained from the first two phases to construct a classifier that works reliably with real–world, unscripted activity data.

In general, this document follows Peffers, Tuunanen, Rothenberger, and Chatterjee (2007) suggestions for how to present research done using the Design Science methodology and in particular the *Demonstration* and *Evaluation* parts of the presentation that Peffers et al. (2007) discuss are included in chapter 5 and chapter 6. As the majority of the evaluation is done within these chapters, the Discussion chapter (chapter 7) for this thesis only covers a summary of the earlier work and a higher level discussion. The *Communication* part mentioned by Peffers et al. (2007) is covered

both by this document itself and by the various peer–reviewed publications that have been published to date, based on this work.

4.2 REASONS FOR CHOOSING DESIGN SCIENCE

The goal within this research is to build a useful artefact (a classifier for punctual movements) that can potentially be used as part of a Wearable Coach concept. As outlined in the introduction to this chapter, Jarvinen (2000) outlines a useful taxonomy and a process for following that taxonomy that, when followed, points to an appropriate research methodology. Within Jarvinen (2000)'s taxonomy the suggested appropriate research methodology when building an artefact is Design Science.

In addition, the author is familiar with Design Science as a research methodology from his prior Masters work and in many ways, this research is a follow on from that prior research that was also done using a Design Science methodology and so it seems somewhat natural to continue using Design Science as the preferred overall research methodology.

It is possible to argue that this work is a "*theory testing approach*" (Jarvinen, 2000, p. 3) because as far as is known no one else has yet used RC methods to build a punctual activity classifier. However, this seems overly pedantic and the review of the relevant literature has shown that while punctual activity classifier research is somewhat ignored as compared to durative activity classifier research, it is, nevertheless, an area of research that has been explored by prior researchers. This makes it hard to argue that there is a "*theory*" that needs to be tested in this area.

Hevner, March, Park, and Ram (2004, p. 3) argue that almost all Information Systems based research is conducted using one of only two research methodologies, Behavioural Science or Design Science:

> *Two paradigms characterize much of the research in the Information Systems discipline: behavioural science and design science. The behavioural science paradigm seeks to develop and verify theories that explain or predict human or organizational behaviour. The design-science paradigm seeks to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts.*

As the goal of this research is to create a *new and innovative artefact* then this work is in good company by choosing Design Science as an appropriate research methodology. In addition the author's prior experience with the Design Science methodology is useful. In this case, other possible methodologies do not seem to be as appropriate. Given these factors, Design Science is then the chosen research methodology for this work.

4.3 DESIGN SCIENCE DESCRIPTION

According to Peffers et al. (2007) there are six steps in the Design Science process and while Peffers et al. list these six steps sequentially they are also very careful to explain that researchers will often not simply move from one step to another in a

linear way but often instead move in a non-linear and iterative fashion between the steps in a manner that suits the particular research process. The six steps described by Peffers et al. (2007) are:

1. **Problem identification and motivation**
   During this step the research question is defined and justified.
2. **Goal setting or objectives of the solution**
   This step outlines the goal(s) of the research and may either be measurable goals such as *X% improvement* or they may be more qualitative such as *A better solution than Y*.
3. **Design and development**
   This is the process where the actual artefact is created.
4. **Demonstration**
   This involves running (or using) the artefact.
5. **Evaluation**
   This step evaluates how well the artefact met (or did not meet) the goals set for it.
6. **Communication**
   This step involves communicating the outcomes of the process to a relevant audience.

As previously described, this research project takes a phased approach and so each of the three phases goes through its own version of the process from step 2 through 6. The design, develop, demonstrate and evaluate cycles that were performed as part of this research are described in Chapters 5 and 6. Sitting over the top of that phased approach is the overall problem identification (Chapters 1 and 2) and the communication of the whole project, represented by this document as a whole plus the various papers that have been published since the research was started. To aid the reader to understand how the six Design Science steps are expressed throughout this document a set of tables 4.1, 4.2, 4.3 and 4.4 on pages 70, 71, 72 and 73 have been provided as a map.

4.4 RESEARCH APPROACH

Chapters 1 and 2 contain the detailed description of the overall research goal for this project along with the justification for those goals, however, for clarity, the overall research goal is summarised here along with the design objectives for each of the three phases. In addition there is a description of the reason for choosing RC as the classifier engine, a brief description of the three phases of development and testing and a list of the design strategies that were followed during this research.

4.4.1 *Motivation for tool selection*

Section 3.2.19 on page 55 describes a number of potential advantages that a RC classifier has over some alternative and more commonly used activity classifiers, par-

Table 4.1: Mapping Design Science steps to this document–Phase 1

	LSM Synthetic Data	LSM Scripted Activities
Problem Identification and Motivation		
Identification	Pg. 95	Pg. 102
Motivation	Pg. 95	Pg. 102
Definition of the objectives for a solution		
Goal	Pg. 95	Pg. 102
Measures	Pg. 95	Pg. 102
Design and Development		
Code	See author	See author
Demonstration		
Description	Pgs. 95–98	Pgs. 102–108
Evaluation		
Results	Pgs. 99–101	Pg. 108
Discussion	Pg. 101	Pgs. 111–112
Communication		
Long form	Pgs. 95–101	Pgs. 102–112
Publications	Schliebs and Hunt (2012)	Schliebs et al. (2013)

ticularly when considering classifying Punctual activities within a wearable device with strongly constrained resources. These potential advantages include:

- The fading memory effect of RC classifiers means that the input sensor data need not be segmented into fixed window sizes and this means that there is potentially more opportunity to successfully classify Punctual activities such as Mounting which exhibit widely varying activity completion time envelopes.
- The fading memory effect may also be advantageous in separating true activities from potential false-positive activities with a "point" similarity in the signal but with signal differences in the lead up to the activity.
- The ability to classify based on relatively unprocessed input means that there is less need for compute resources during classification and this has benefits when implemented on a resource restricted wearable device and potentially speeds up the classification process.
- The ability to use unsegmented input means that the classifier can classify on a sample by sample basis and so this reduces the lag time between waiting for the next segment window to close and this potentially speeds up the classification process. This potential speed up has positive implications when using the classification to drive a coaching response in real-time.
- The resource usage profile of RC methods make them particularly suitable for implementing in a resource constrained wearable device situation. RC classifiers often require considerable resources to train and tune but once trained they require considerably less resources to do the actual classification of new data.
- The RC engine's ability to classify multiple activities from the same reservoir has potential benefits in both reducing the storage requirement and enabling

Table 4.2: Mapping Design Science steps to this document–Phase 2

	Improve Performance (ESN & PSO)	**Improved Classes**
Problem Identification and Motivation		
Identification	Pg. 113	Pg. 130
Motivation	Pg. 113	Pg. 130
Definition of the objectives for a solution		
Goal	Pg. 113	Pg. 130
Measures	Pg. 114	Pg. 130
Design and Development		
Code	See author	See author
Demonstration		
Description	Pgs. 114–117	Pgs. 130–132
Evaluation		
Results	Pgs. 117–119	Pgs. 133–140
Discussion	Pgs. 119–126	Pgs. 140–141
Communication		
Long form	Pgs. 113–126	Pgs. 130–141
Publications	Hunt et al. (2014)	

the parallelisation of the classification process. Both of these potential benefits could be significant in a wearable device environment.

- Some researchers such as (Sheik et al., 2012) have been able to implement similar RNN architectures directly in hardware and so this is also a potential benefit in a wearable environment with constrained resources.
- RC classification engines have been used successfully in related fields with signals with a significant temporal aspect such as isolated word recognition (Verstraeten et al., 2005), speech recognition (Jaeger et al., 2007; Verstraeten et al., 2006) and spoken phoneme recognition (Goodman & Ventura, 2006) and so there is a good chance that this tool will also work with spatio-temporal inertial data. In addition, (Verstraeten et al., 2007) argues strongly for RNN's as a powerful, general purpose, temporal machine learning tool.
- The use of RC classification engine for human activity classification is novel and so this helps to fulfil the requirement to produce a novel and innovative artefact as part of the Design Science methodology.

Having listed the author's reasons for choosing this particular tool for use within this research and given that the Design Science research goal is to produce a useful artefact there is no particular reason to test the ability of a RC classification engine to meet these benefits at this stage. The primary role of this research is to produce a working artefact that meets the design requirements. Once this process has been achieved then there is an opportunity in follow on research to test RC classification techniques against other, more commonly used classification techniques to see if RC techniques have a real benefit when classifying Punctual human activities within a resource constrained wearable device environment.

Table 4.3: Mapping Design Science steps–Real World Data, Part 1

	Initial	**Offset**	**Ensemble 1**	**Ensemble 1A**	
Problem Identification and Motivation					
Identification	Pg. 149	Pg. 160	Pg. 169	Pg. 180	
Motivation	Pg. 150	Pg. 160	Pg. 169	Pg. 180	
Definition of the objectives for a solution					
Goal	Pg. 150	Pg. 160	Pg. 169	Pg. 180	
Measures	Pg. 150	Pg. 160	Pg. 170	Pg. 180	
Design and Development					
Code	See author	See author	See author	See author	
Demonstration					
Description	Pgs. 150–154	Pgs. 160–163	Pgs. 170–173	Pgs. 180–182	
Evaluation					
Results	Pgs. 154–159	Pgs. 163–168	Pgs. 173–178	Pgs. 182–187	
Discussion	Pg. 159	Pg. 168	Pg. 178	Pgs. 187–188	
Communication					
Long form	Pgs. 149–159	Pgs. 160–168	Pgs. 169–178	Pgs. 180–188	
Publications	Hunt and Parry (2015)				

4.4.2 *Overall Goal*

The overall design goal for this research is the construction of an ACS to classify punctual activities of interest within Equestrian Sport and in particular, to construct a working system using real or realistic data so that a RC based punctual ACS could become a useful component of a "wearable coaching system" for equestrian sportspeople. The artefact produced as part of the design science process is the automatic classifier. This could form part of a larger coaching system in the future.

4.4.3 *Goal setting and research framework for each phase*

The approach used to achieve the overall goal is to break the design, build and test process down into three separate phases. This allows for different design measures for each phase and as progress is made from phase to phase, this allows for some confidence that progress is being made. The three phases of this research are a proof–of–concept phase; a phase where a classifier is built using data from scripted, laboratory based equestrian activities and the final phase where a classifier is built using real world data from unscripted equestrian activities. Each phase has its own set of goals and within each phase, the particular design goals for that phase are discussed, along with any activity descriptions that are local to that phase. The goals form the high level performance measures against which the artefacts developed during the research are measured. The research objectives are described in terms of classification and are inclusive of any data pre-processing required to enable more reliable classification, especially when classifying across subjects and places in the real world. In addition, the research framework used within each phase is also described within the description for that phase. The design goals for each phase are summarised here.

Table 4.4: Mapping Design Science steps–Real World Data, Part 2

	Under-sample	**Butterworth**	**Accelerometer**	**Lukosevicius**
Problem Identification and Motivation				
Identification	Pg. 190	Pg. 199	Pg. 207	Pg. 224
Motivation	Pg. 191	Pg. 199	Pg. 207	Pg. 224
Definition of the objectives for a solution				
Goal	Pg. 191	Pg. 199	Pg. 207	Pg. 224
Measures	Pg. 191	Pg. 199	Pg. 208	Pg. 224
Design and Development				
Code	See author	See author	See author	See author
Demonstration				
Description	Pgs. 191–192	Pgs. 200–201	Pgs. 208–209	Pgs. 224–226
Evaluation				
Results	Pgs. 192–197	Pgs. 201–205	Pgs. 209–221	Pgs. 226–231
Discussion	Pg. 197	Pg. 205	Pgs. 221–222	Pg. 231
Communication				
Long form	Pgs. 190–197	Pgs. 199–205	Pgs. 206–222	Pgs. 224–231
Publications	To be submitted			

Phase 1: *A proof of concept RC based classifier that is capable of classifying complex spatio-temporal data.*

This work culminated in the publication of Schliebs and Hunt (2012) then Schliebs et al. (2013) and is described in Section 5.2. During this proof–of–concept phase a LSM was used to classify the activities of interest.

Phase 2: *A RC based classifier that is capable of classifying non-windowed, scripted punctual activities with reasonably consistent temporal activity frames across limited subjects.*

During phase 2 a secondary objective was to build a Researcher's Workbench for RC punctual activity classifiers. This work is described in Section 5.4 and was reported in Hunt et al. (2014). During this second phase the RC model was changed from a LSM model to an ESN model for classifying activities for ease of use and for performance improvements during the search of the parameter space associated with each RC method.

Phase 3: *A RC based classifier that is capable of classifying non-windowed, real world, punctual activities with possibly divergent temporal activity frames across multiple subjects in differing geographic locations and on different session dates.*

This work has its own chapter, namely Chapter 6 and the initial results were reported in Hunt and Parry (2015). Another paper is envisaged that will report the final results from this phase and, of course, this thesis also reports the full results from this phase.

4.4.4 *Design Strategies*

This work has been directed by a small set of Design Strategies, these strategies are:

- **Favour simplicity** - Given a choice between a simple method or a more complex one then favour the simple one.
- **Design for usability** - When making design choices, prefer methods that will make the software artefact easier to use.
- **Be mindful of possible implementation issues** - Keep in mind possible implementation requirements without being absolutely constrained by them
- **Treat the data as a black box** - This work does not attempt to map the raw data into real world orientation or movement coordinates, it is simply either processed as it was received or simple features are calculated from it. Jaeger and Haas(2004) suggests that ESN's are ideal for Black Box classifiers. Newer sensors, such as YOST Labs (2015), are capable of outputting quaternions and other such interpretations of the raw data and so using data from such sensors may provide improved classification in future but in this work this has been excluded so as to control the scope of the work (*favour simplicity*).

  The signal to noise ratio is low for magnetometer data especially and for all three sensors in general but the data available is the data available. A number of months were spent investigating noise reduction techniques but a decision was then made that this was unnecessary at this stage as newer sensors, such as YOST Labs (2015), are being built so that they filter noise internally (within the sensor) in many cases. An algorithm built to use the noisy data that is available for this research should be transferable to data with less noise, albeit with perhaps some changes.
- **Tailor the approach to the known data characteristics** - The Magnetometer uses a real-world frame of reference and so data is dependent on where the activity takes place and what direction the rider and horse are facing as the activity occurs. As a result initial classification of the scripted, laboratory-based activities (which all took place at the same position within the laboratory using the same orientations) will include Magnetometer data so that its value can be evaluated but final classification of the scripted activities and all real-world activities will not use the Magnetometer data.

  At the beginning of the design iteration process two datasets of scripted activities will be used to gain a deeper understanding of some of the design elements that underlie classifying this type of activity using this type of data. The scripted data is a useful place to start because the activities of interest are much more common within these datasets than they typically would be in a real life dataset. In real life, a rider would typically mount and dismount only once or twice during a training session, whereas in these datasets there are 17 and 40 mounts and dismounts. In addition, the regularity of action imposed by the activity script ensures that there are or should be regular patterns in the data that will hopefully lead to some early successes to help drive the design

iteration process. Lastly, having two different datasets from two different participants with differing heights, strength and mounting/dismounting style who also have slightly different interpretations of the the script gives (hopefully) enough differences that classification is non-trivial.

Once there start to be some successes, however, the data will be swapped to the real-life data so that classification can move beyond the artificial patterns within the data that come from the regular en-action of a scripted activity.

4.5 ACTIVITIES OF INTEREST

Within this work, the punctual activities of interest that are the focus of the classifier are mounting and dismounting a horse. These activities are defined in Chapter 2, Section 2.2. These activities mark the boundaries of "riding" and so successful classification of these activities will be consistent with the Design for Usability strategy in that successful classification of these activities will allow for the automatic detection of the start and end of riding. Being able to automatically detect the start and end of riding provides important context to a Wearable Coach and enables riding coaching to (automatically) start and end with riding.

Thirdly, these activities are interesting and useful because they:

- are punctual activities
- they are complex enough to be interesting while (hopefully) not so complex that classification is improbable
- there is some level of standardisation of technique across riders that comes from strong traditions (mount and dismount from the horse's left side) and commonality of equipment and function (need to get on and off this horse that is usually much taller than a rider's hip using a stirrup and saddle while ensuring that it stands reasonably still)
- there is noticeable temporal variation in the time taken to do these activities by different riders in different real–world situations. As an example, in a sample of 55 real life mounts, the mean time to mount was 5.48 seconds, with a standard variation of 2.28 seconds, a minimum of 1 second and a maximum of 10 seconds (Hunt, 2009, pp. 125).
- there are other variations in the real world situation as a result of different levels of strength, athleticism, suppleness, skill, height and attitude of the rider and height, level of training and personality of the horse.

Lastly, based on the prior Masters project and domain experience, it is thought that there is a reasonable chance that a rider's individual arm movements during these activities are not totally unique within riding while the combined effect of the movements as measured by the sensor leading up to and across the totality of these activities might be unique enough to successfully classify without being overwhelmed by false positives.

4.6 DATA

This section describes the data used in this research. A constraint placed on this research by the author's university was that no additional data was to be captured. As a result this research uses a subset of the data collected as part of the author's Masters project, Hunt (2009). The author's Masters project was essentially a data capture exercise with some simple analysis of the data. The Master's thesis, Hunt (2009), is a good source for detailed descriptions of the data capture process and equipment, curious readers are referred to that document for these details.

The data was captured under guidelines agreed in ethical approval reference number AUTEC 08/47 as approved by the Auckland University of Technology Ethics Committee on 24th April 2008. All participants were asked to consent to the use of the data captured from their participation sessions and all participants signed a release of copyright for the use of their images in video and still formats. The consent and release form was presented in both English and Swedish. Copies of the ethics forms can be found in Hunt (2009, p.191–210).

At the time of data capture no decision had been made on how the data would be classified and so only very basic data preparation and analysis was done during the Masters project. During this work the required data preparation was completely redone, based on the chosen classification method. The raw data was carried over from the author's Masters project but no data preparation or data analysis was carried over.

The collection of datasets used within this research consists of both scripted (laboratory) and unscripted (real world) data. The scripted data includes two datasets collected in a laboratory situation using a wooden *horse* and two different participants. The real world (unscripted) data consists of 10 datasets collected from 7 different participants across 9 data collection sessions. There are two files for three participants and a single file for four participants. The laboratory collection of datasets contains scripted mounts and dismounts plus moving between the two activities. The real world collection of datasets cover a wide variety of unscripted equestrian related activities including catching horses, grooming horses, tacking up, mounting, riding (including standing, walking, trotting, cantering, jumping and turns), dismounting and untacking horses.

All inertial data was captured by a single IMU sensor worn on participants right wrist and was captured at 10Hz and the reasons for choosing this data rate are set out on page 87 of Hunt (2009). Given that this research was constrained to using the existing data from the Masters project, there was no option to capture data at a higher frequency. However, Verplaetse (1996) demonstrated that the human hand and lower forearm moves with a frequency of less than 12Hz and so the 10Hz frequency of data capture is certainly in this ballpark. In addition, during early data preparation for the development of this classification artefact we down sampled the data to 5Hz by taking two streams of every second sample value and when these data streams were plotted during horse mount sequences no real difference was noted between the 10Hz and 5Hz data. It is also worth noting that the purpose of this work is to capture and analyse data for **classification** purposes rather than for some detailed technique or style

analysis or to generate some sort of bio-mechanical model of movement and so the lower data capture rate is consistent with this purpose, especially given Verplaetse's work and our own desire to develop a classification artefact that is capable of classifying quickly. Unless there is additional information value, sampling at higher rates simply imposes a much greater computational demand on the classifier.

As noted in Hunt's Masters thesis a data capture rate of 25Hz would have closely matched the video frame rate for the video camera and would have made synchronising the inertial and video data much simpler and the additional data from a higher sampling rate may hold additional data value. The potential trade-offs between a higher data capture rate and possible added data value could be tested in some future work.

4.6.1 *Summary of Laboratory Data Capture Sessions*

There are two sensor files from laboratory data capture sessions, 2008081301SF_LA (LALab) and 2008082102SF_LB (LBLab). LALab was the first laboratory data capture session and some lessons were learned during this session that were then applied for the second, LBLab, session. The start of the file contains a very long sequence of no or little action as the IMU was recording data as the author set up the rest of the equipment such as the tripod mounted camera and explained the session script requirements to the participant. For the second laboratory data capture session the equipment set up and explanation were done before turning on the IMU and so the mounts and dismounts start almost immediately.

LA was a more cautious participant than LB and so she followed the session script in a less exuberant manner than LB, who used a lot more energy during his session. This contrast in exuberance led to LB doing a lot more mounts and dismounts than LA (40 versus 17) and in conjunction with a different dismount technique meant that LB abandoned the mounting block that was supplied to both participants after the first dismount when his legs missed the mounting block on dismount and he almost twisted his ankle. LA, on the other hand followed a more measured approach while dismounting and always found the mounting block. This change by LB meant that his first mount and dismount are different from his subsequent mounts and dismounts as a result of stopping using the mounting block and the almost fall during the first dismount. In addition to the difference of use and non-use of the mounting block, LB's dismount technique is different from LA's and this shows up in the data. LB dismounts by sliding off the saddle quickly with no feet in the stirrups while LA lowers herself with her left foot still in the stirrup until her right foot is on the mounting block.

In addition, LB has a greater variance in mount and dismount times as the session progresses with a couple of cautious mounts and dismounts followed by quicker mounts and dismounts as he gained confidence, tapering off to slower mounts and dismounts (further apart and harder to see, but, longer intervals for a mount and dismount) as he grew tired from the effort of mounting and dismounting and then a quick burst for the last two mounts and dismounts. LA's mounts and dismounts, on

the other hand, are more regular as she paced herself and stopped before she got too tired.

4.6.2 *Summary of Real World Data Capture Sessions*

The real world data capture sessions that were used in this research are summarised in the following table.

Date	Venue	Session	Notes
0716	Tierp	RA2	Pre-riding and mount
0719	Mariefred	RB1	Short pre-riding but long walk at end after dismount
0812	Vallentuna	RA2	Pre-riding data including brushing
0829	Bro	RC2	Short pre-riding but long walk at end after dismount
0830		RD3	Pre and post-riding
0902	Bro	RE1	Pre and post-riding
		RC2	Short pre and post-riding
0912	Orebro	RF1	Long pre and post-riding
		RG2	Long pre-riding and no post-riding
0913	Orebro	RF3	Long pre and post-riding

Table 4.5: Summary of used real world data

4.6.3 *Synchronising the Data*

The general process for synchronising the video and sensor data and then generating class vectors for activities of interest is shown if Figure 4.1. The sensor data files need to be synchronised with their associated video files so that activities can be identified on the video and then marked in a class field that is added to the sensor file.

Each data capture session started with the participant being asked to perform a synchronisation activity. In some cases, the participant was also asked to perform the synchronisation activity at the end of the session. These cases allowed for a check for drift between the sensor data and the video frame rate.

Synchronisation then consisted of viewing the video stream, frame by frame to find a known point within the synchronisation activity (top-of-stroke for arm waves and moment-of-impact for overhead claps). Then viewing a plot of the sensor data to find the probable equivalent point, see Figure 4.2. The repetition of the activity made it a lot simpler to find the probable point within the sensor data because the produced pattern (as plotted) was unusual and repetitive. Having found the probable point of synchronisation, (in this case the top of the Gyroscope Pitch axis) this was then tested against some other easy to spot pattern such as an end of session synchronisation activity or a significant period when the participant had their arm still for some reason (see Figure 4.5).

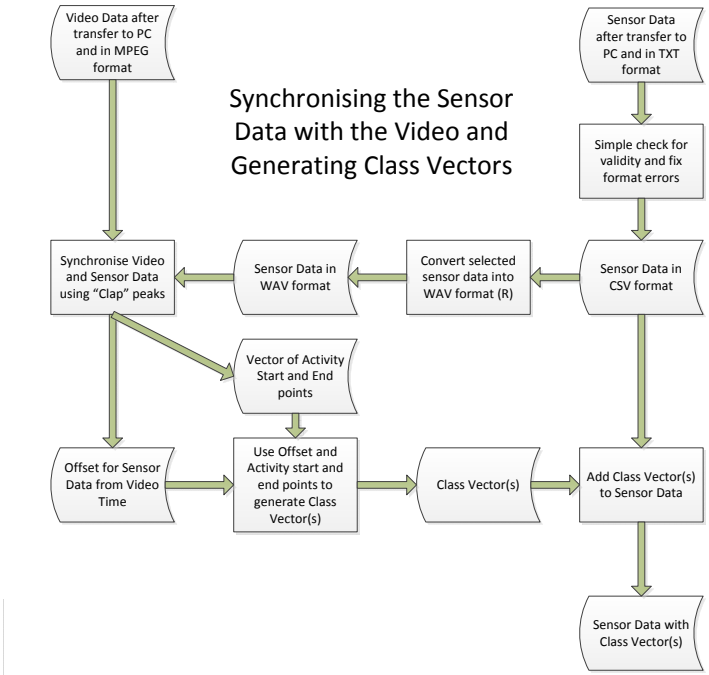


Figure 4.1: Process for Synchronising and Generating Class Vectors

Two screen-shots (Figure 4.3 and 4.4) illustrate this process. The first screen-shot (Figure 4.3) of Vegas Movie Studio shows a (time) zoomed in view of the video file for session 2008091202RG. The participant can be seen clapping her hands over her head. Below the larger picture are five editor streams. The first shows, at this time resolution, a frame by frame depiction of the video. The second stream, entitled voice in the screen shot and coloured purple, shows the sound recorded with the video. The clap sound can be clearly seen within this stream, immediately to the left of the cursor (and associated white vertical line).

The next three streams below this are the Accelerometer and Gyroscope sensor data synthesized into MS Windows .wav files and imported into the video editor. The video and sound streams are then slid until they match the likely equivalent position in the sensor streams. In this case the trailing edge of the frame with the clap sound within it is matched with a peak from the Gyroscope Pitch axis. This can be seen a bit more clearly when the time dimension is zoomed out a little at the same point within the video, see figure 4.4.

In this case there is also a synchronisation sequence at the end of the video (Figure 4.8) and so this is used the same way to test both that the synchronisation point at the start is valid and that there is no real drift between the video stream and the sensor stream.

A simple spreadsheet is used to record the frame numbers of the synchronisation points and the activities of interest from the video. Then the offset between the video and sensor data is calculated using the synchronisation point at the start. The prob-

able sensor data sample number for the synchronisation point at the end can then be calculated, that general area is then plotted to ensure that there is a match.

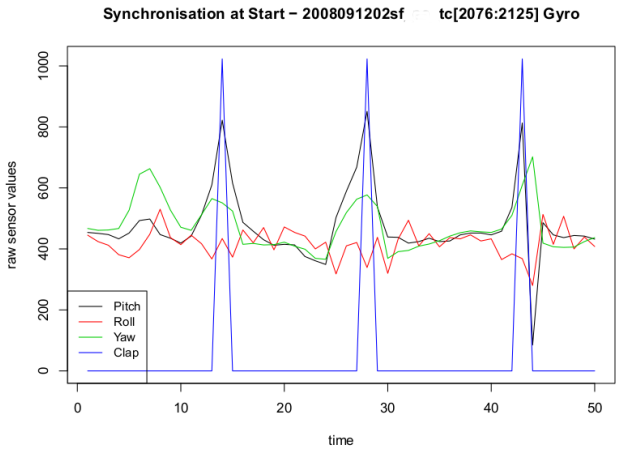


Figure 4.2: Identifying the Start Synchronisation Point for 2008091202RG

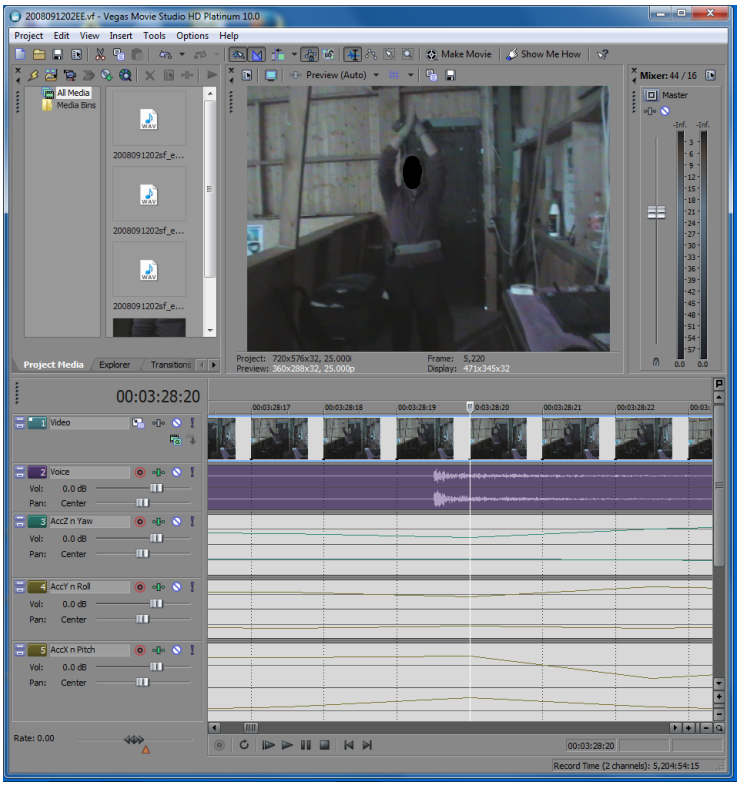


Figure 4.3: Identifying the Synchronisation Point for 2008091202RG – Zoomed In

Once there is reasonable surety that the video and data are synchronised then the mount and dismount ranges are calculated, plotted for a visual check (see Figure 4.6 and 4.7) and a class column is added to the sensor data file with this information within it.

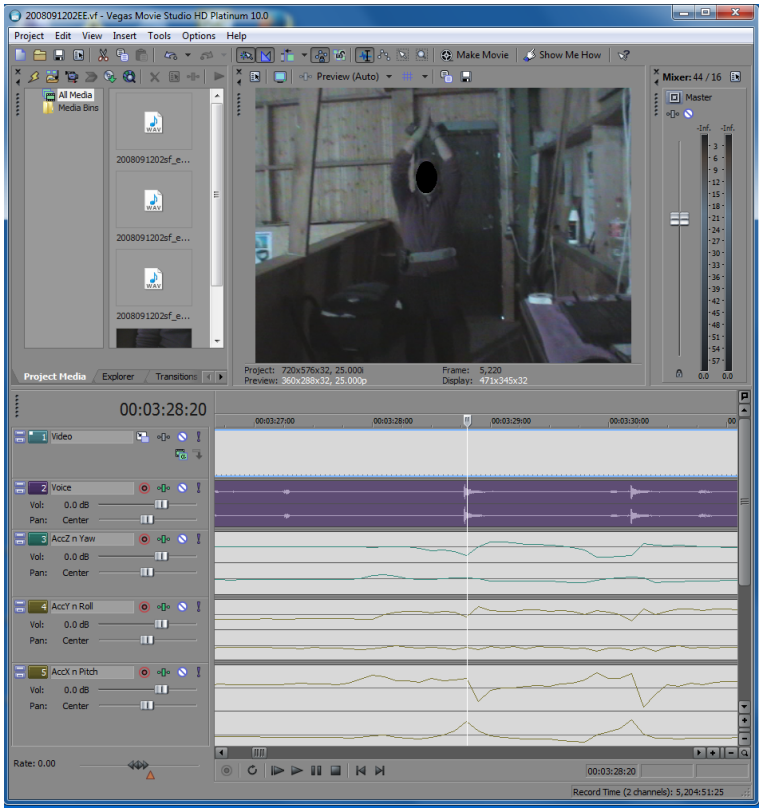


Figure 4.4: Identifying the Synchronisation Point for 2008091202RG – Zoomed out

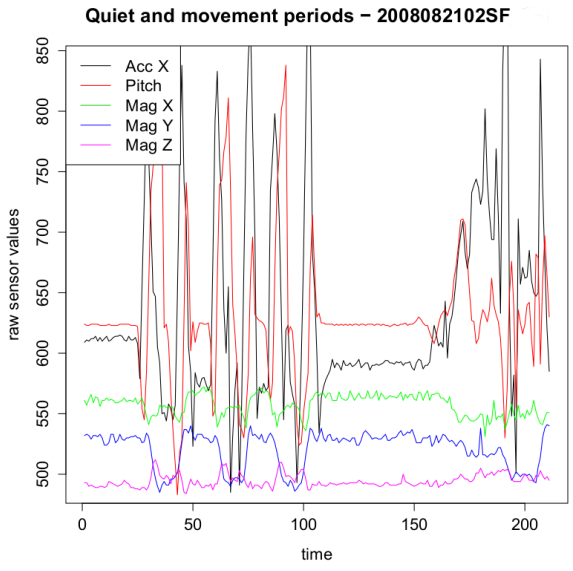


Figure 4.5: Significant Quiet and Movement Sequences for 2008082102LB

The two outputs of the synchronisation activity are a spreadsheet per session that documents the point of synchronisation within both the sensor data and video files

with the timing offset between the files and the updated sensor file with the class column populated within it. The offset is important as it allows the activity in the video to be identified and that allows the equivalent data for that activity to be identified within the sensor data.

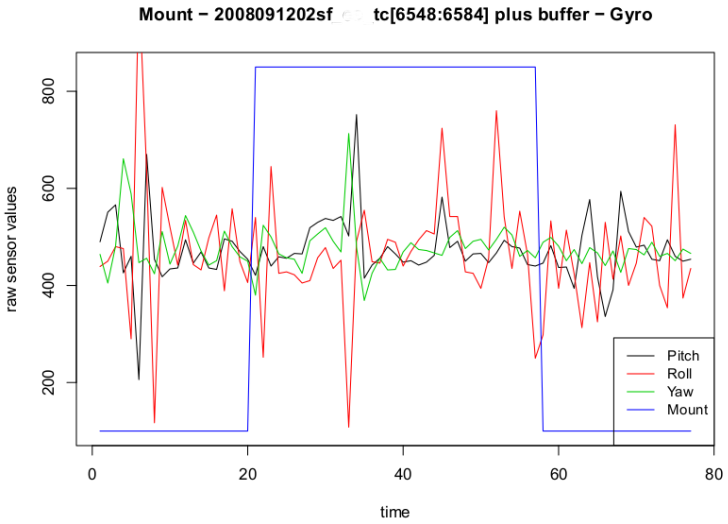


Figure 4.6: Mount Segment from 2008091202RG

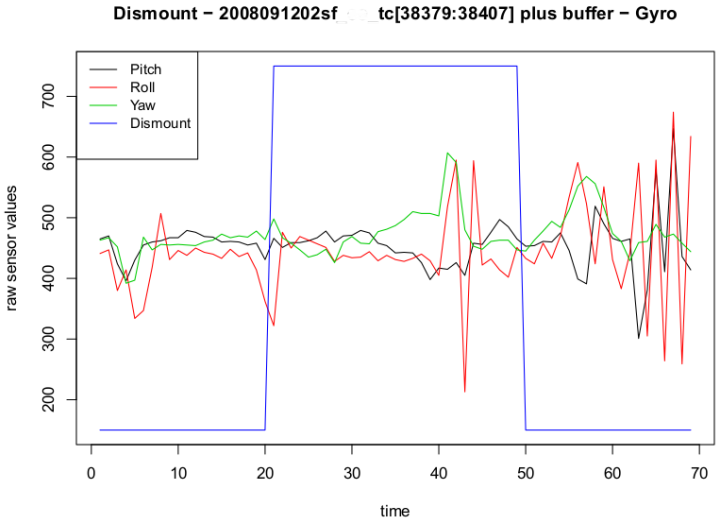


Figure 4.7: Dismount Segment from 2008091202RG

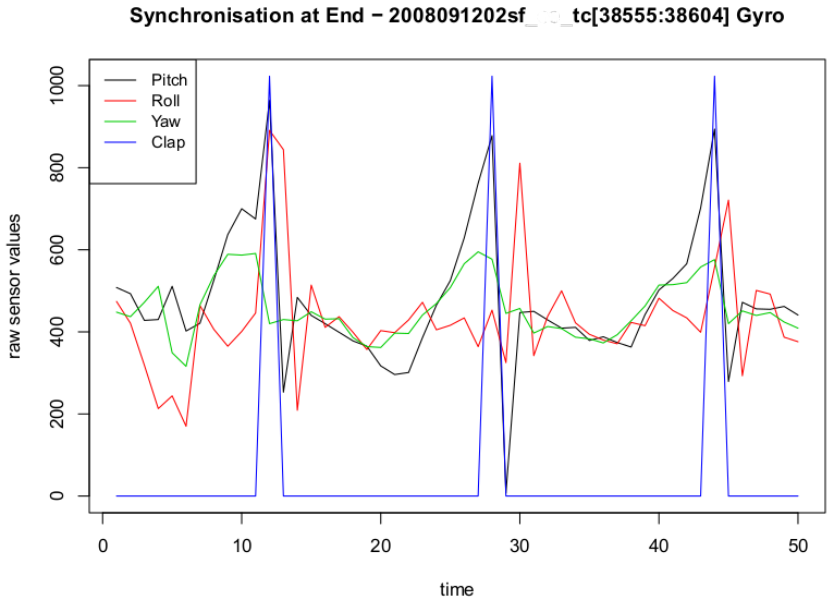


Figure 4.8: Identifying the End Synchronisation Point for 2008091202RG

Description	Video Time	Video Seconds	Data ID#	Data Seq#	Data Seconds
Start	0:00:00:00	0.00	1,054	0	0.0
End	1:06:35:05	3995.20	48,539	47,486	4748.6
Sync Point	0:00:02:14	2.60	7,808	6,755	675.5
Offset from Video			+7, 782	+6, 729	+672.9
Mount Start	0:11:55:16	715.64	14,938	13,885	1388.5
Mount End	0:11:59:18	719.72	14,979	13,926	1392.6
Mount Elapsed		**4.08**			**4.1**
Dismount Start	0:58:32:24	3512.96	42,911	41,858	4185.8
Dismount End	0:58:36:22	3516.88	42,950	41,897	4189.7
Dismount Elapsed		**3.92**			**3.9**

Table 4.6: Synchronisation for 20080717RA2

4.6.4 *Data Issues & Characteristics of the Sensor*

At this point it is useful to discuss the characteristics of the IMU sensor that was used to capture the data and to review its components. Soon after purchasing the SparkFun® 6 DoF V4 IMU it was discovered SparkFun Electronics Inc (2008b) that the sensor has a design fault that causes the BlueTooth® radio to interfere with the Magnetometer. The Magnetometer signal as generated by this sensor module, contains at least three non-random noise effects that are caused by the BlueTooth® radio outputting electromagnetic radio waves that are picked up by the Magnetometer.

The first effect is caused by the intermittent nature of the BlueTooth® send and receive protocol. Radio waves are sent and received in burst mode. This means

that the radio turns on and off at a very high frequency (2.4GHz), much higher by a number of powers than the frequency at which the sensor was set to sample the Magnetometer for human movement data (10Hz).

The second effect is caused by the nature of the power conservation and error correction protocols. Bluetooth® is a low-power, short-range radio standard and when both the send and receive stations are close to each other (and not masked) then the protocol reduces the amplitude of the signal so as to conserve power, while at greater distances, when more errors are encountered the signal amplitude is increased to reduce the error rate. This means that with the receiver in a fixed position on the participant's hip and with the transmitter on their wrist then as they move their arm closer to or further away from their body then the radio signal amplitude decreases and increases. This effect happens at the expected frequency of human movement.

Thirdly, the Bluetooth radio frequency is such that it is easily absorbed by water. The human (and horse) body are primarily composed of water and so this means that when the participant's movement is such that when the wrist is partially or completely occluded by a body part then again the signal amplitude is increased. The frequency of this effect is generally less than the frequency of human movement.

In addition, the Magnetometer also picks up hard-iron magnetic interference from ferrous metals close to the sensor and electromagnetic interference from other radio sources in the immediate environment. During mounting a rider often places their hand close to the stirrup (often iron) to steady it, just prior to mounting and there are often other iron items such as buckles around the saddle and bridle. Often there are other much larger ferrous objects around where horses are kept and ridden such as iron gates and iron door supports. Mobile phones are becoming ubiquitous and they produce radio waves in burst mode so that mobile phones that are reasonably close at hand will also generate interference.

All of these sources of interference are picked up by the Magnetometer and are seen as non-random noise at differing frequencies, including from a very high frequency (bursty traffic), a frequency consistent with the sample rate (movement associated effects) and a much lower frequency than the sample rate (occlusion effects). While all of these interference sources are known issues with Magnetometers this particular sensor module suffers interference much more than normal as a result of its design (SparkFun Electronics Inc (2008b)).

Another design decision makes this even worse. The sensitivity of the Magnetometer in this device is permanently set so that the Earth's magnetic field readings only cover a small, central subsection of the possible data range. Looking at the 22 data files with right arm sensor usage it is possible to see (Table 4.7) that the standard deviation for the individual magnetometer axis is quite low compared with the Gyroscope and very low compared with the Accelerometer. With the data values pushed into this central region and with the presence of non-random noise the signal to noise level is quite low.

Calculating the mean and standard deviation across all 12 files used in this research and then for the example, sample area with still and large movement segments from LBLab (shown in Table 4.7), it is possible to see that the Magnetometer axis show the least gross variation overall and that for the example area (Table 4.8) the relatively

still segments standard deviation is proportionally large compared to the large movement segments standard deviation. From this it can be concluded that other authors (SparkFun Electronics Inc (2008b)) are correct and that noise does tend to dominate the Magnetometer signal.

Table 4.7: Mean and Standard Deviation for Raw Signal Data

Sensor	Mean	Std D
Magnetometer X	557.1	16.0
Magnetometer Y	514.5	18.0
Magnetometer Z	506.5	15.3
Accelerometer X	623.7	130.7
Accelerometer Y	455.7	119.2
Accelerometer Z	560.4	102.6
Gyroscope Pitch	471.0	36.6
Gyroscope Roll	465.0	72.5
Gyroscope Yaw	470.4	33.7

Table 4.8: Variance for a Sample Region with Still & Movement Data from LBLab.

Sensor	Movement SD	Still SD	Overall SD
Magnetometer X	7.7	2.9	16.0
Magnetometer Y	14.4	3.0	18.0
Magnetometer Z	5.1	1.9	15.3
Accelerometer X	89.4	10.5	130.7
Accelerometer Y	101.3	4.4	119.2
Accelerometer Z	86.0	9.9	102.6
Gyroscope Pitch	47.7	0.6	36.6
Gyroscope Roll	57.9	1.0	72.5
Gyroscope Yaw	59.2	0.5	33.7

Figures 4.9, 4.11 and 4.13 show histograms of the raw sensor data for each sensor. In addition, figures 4.10 4.12 and 4.14 show density functions for the same raw data. This enables slightly easier visual comparisons.

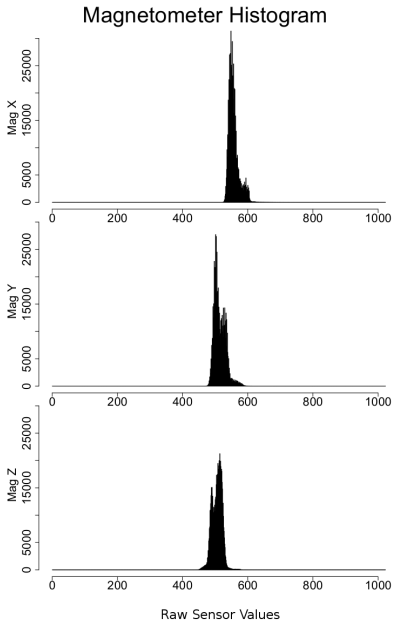


Figure 4.9: Histogram of Magnetometer Raw Values across All Files

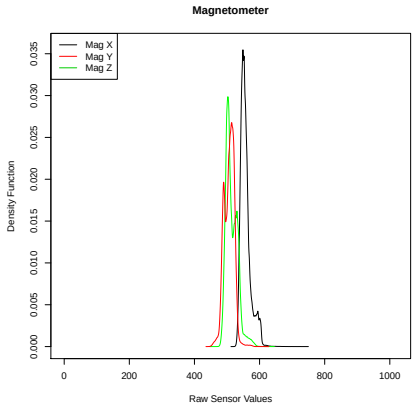


Figure 4.10: Magnetometer Density Function across All Files

The Magnetometer histograms and Density plot show graphically that the signal from this sensor covers only a narrow, central'ish region of the possible signal range. Ideally, when this data is transformed for input into the ESN model, the transformation algorithm would use something like $X_{norm} = \frac{X - \min(X)}{\max(X) - \min(X)}$ to normalise the data within the range of 0 to 1 so that the Magnetometer data is able to influence the ESN model with similar weights to data from the other sensors. If, however, this same transformation is used on the Magnetometer data then not only does this amplify the data in the signal but it also amplifies the noise.

While some authors such as Jaeger (2005) suggest that adding noise to the inputs into an ESN can improve the model's ability to generalise, they are more specific and suggest that the added noise should be both random and a relatively small proportion of the signal. In the case of the Magnetometer data the noise is neither random nor proportionally small.

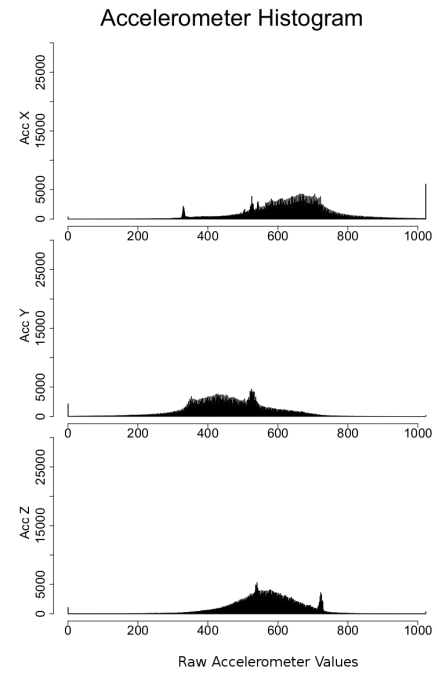


Figure 4.11: Histogram of Accelerometer Raw Values across All Files

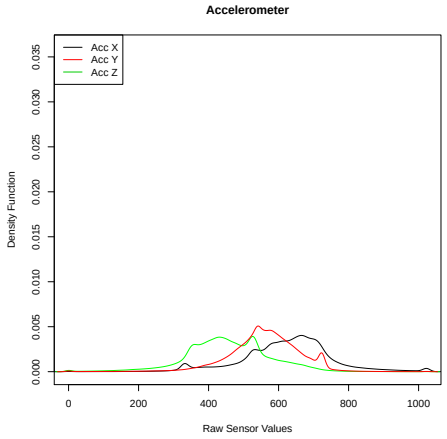


Figure 4.12: Accelerometer Density Function across All Files

The Accelerometer histograms and density plot are almost the opposite of the Magnetometer with a broad spread across the whole range and with some notable peaks at zero, 1023 and at other points across the range. The general, multi-modal type distribution for the Accelerometer signals are generated by the mixture of acceleration sensed from movement and acceleration sensed from Gravity.

It is important to keep this multi-modal tendency in mind when considering using mean based transformations when the data is pre-processed as the means will be skewed by the different peaks.

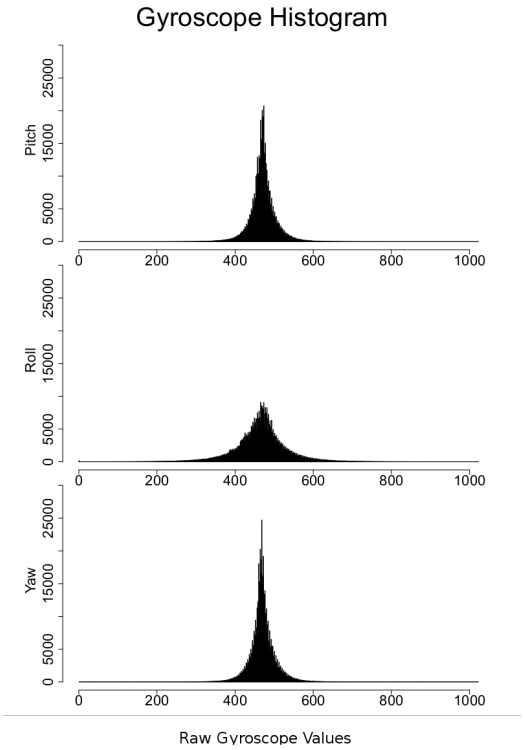


Figure 4.13: Histogram of Gyroscope Raw Values across All Files

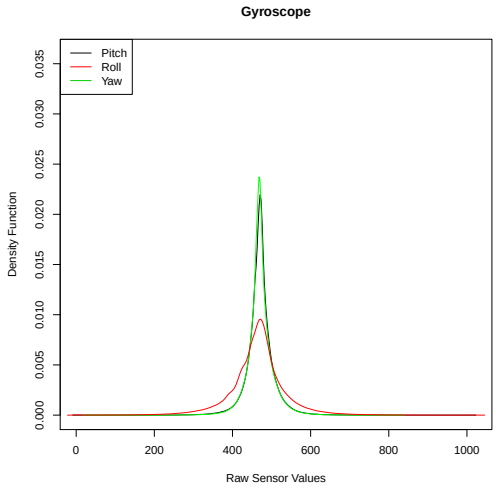


Figure 4.14: Gyroscope Density Function across All Files

The Gyroscope histograms and density plot demonstrate that this sensors signal covers the full range with a central peak. This central peak lines up quite well across all three axis. This data looks the cleanest of the three and as was seen earlier there seems to be little noise pollution of this data.

The potential downside of this tendency to a strong central peak is that when this data is Normalised and then transformed into a -1 to +1 range then much of the

data will cluster around zero and Jaeger (2005) suggests that when using a sigmoid function within the ESN that it may not be a good idea to have data that is strongly clustered around zero as at that point a sigmoid function produces an almost linear effect.

4.6.5 *The Black Box approach and the Magnetometer Data*

The Black Box approach to data within this research means that no attempt is made to understand or take meaning out of the data but instead it is simply used as is, albeit with some transformation to assist with comparability.

This represents a challenge when an attempt is made to use the Magnetometer data as unprocessed Magnetometer data as it is location and orientation specific. For example, If a person is facing North and then turns left prior to mounting their horse then the raw magnetometer data is different from if they start off facing south and then turn left prior to mounting.

Further, someone in Stockholm, Sweden facing North prior to turning left will generate a different signal from someone in Auckland, New Zealand who starts by facing North prior to turning left.

While there are algorithms that could be applied to the Magnetometer data to turn it into something that enables a left turn to be recognised regardless of initial orientation and regardless of location, this would break the Black Box approach.

This plus the issues of noise within the Magnetometer data from this sensor resulted in a decision to exclude the Magnetometer data from all except the very initial trials in this research.

4.6.6 *Splitting the Data into Training and Testing Sets*

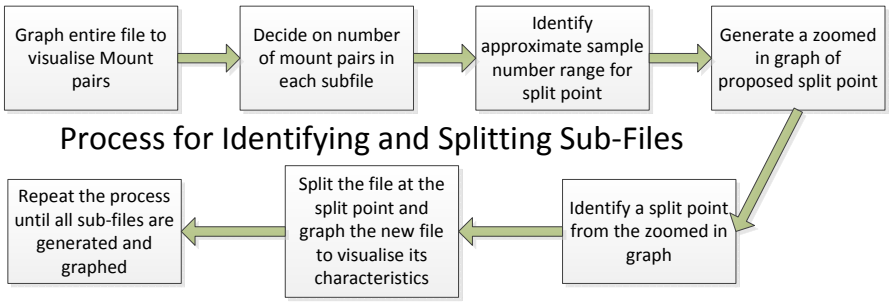


Figure 4.15: Process for Splitting Lab Files

While it was not necessary to split any real world data files it was necessary to split the laboratory files into separate sub-files because each laboratory file contains multiple mounts and dismounts and there was a need to ensure that the mounts and dismounts

in the test data were kept completely separate from the mounts and dismounts in the training data. It is useful to have multiple files so as to keep tuning data separate from training and test data whenever possible but there was also a need to have enough examples of an activity to successfully train the models. In addition, when combining data from the two participants to test for more generalised activity classification (beyond successful classification of activities within a single participant) then it is also useful to have similar numbers of example activities in the training set, although the test set need not be so balanced. The general process of splitting the laboratory files is described in Figure 4.15.

The earliest, tests using the Laboratory data within an ESN made use of a truncated part of the file 2008081301LA. In this early work data was deliberately cut out of the data at the beginning of the file that did not show mounts and dismounts and then the remaining data was split in half without regard for where the split took place.

The thinking in these early stages was that the data at the start of this file was somewhat irrelevant as it is not typical of any equestrian sports activity but is instead related to the participant putting the sensor on their wrist and then standing around relatively still as other items of equipment were set up.

Work with this subset file tended to show up false positives during the non-mount/dismount period towards the end of the file when the participant was taking off the sensor. On further thought it was decided to either include the whole file with the expectation that by including additional non-mount/dismount data during training there would be a better chance of correctly classifying the trailing non-mount/dismount data or else delete the trailing non-mount/dismount data.

It was decided that it was more logical to include both the leading and trailing data rather than risking suggestions of deliberate manipulation of the data by taking out selected data. Subsequently, it was decided to split the full LALab (with 17 mount/dismount pairs) into two files, one containing 9 and the other 8 mount/dismount pairs. This was useful for situations where a model was only trained and tested but once separate optimisation tests were run to preselect a set of parameters for the ESN then four files, rather than two were needed, so that it was possible to keep the training and testing of the end model independent of the data that was used to tune the ESN parameters.

In addition, it was also important to ensure that during training of the model for two participant tests that there were similar numbers of training mounts from each participant so that the data did not bias the training towards a particular participant. This was found to be particularly important for the dismounts because each laboratory participant used a slightly different method of dismounting that showed up in early trials.

LBLab contains 40 mount/dismount pairs and so a natural split for this file was into four sets of 10 mounts and dismounts. However, splitting LALab into four similar sized files would yield three files of four and one file of five mounts/dismounts and this number risked being overwhelmed by the more than twice as many LBLab mounts and dismounts.

As a compromise between keeping the data independent and balancing the number of mounts and dismounts from each participant it was decided to make two further

splits in the LALab file. One to produce a 10 and 7 split and the other to produce a 7 and 10 split. With two sets of LALab sub-files split at different points it was possible to then use similar numbers of mount/dismount pairs during training (e.g. first 10 during tuning training and last 10 during final trials) with some independence between tuning and final trials at the expense of some re-use of data.

The reservoir of the ESN starts in an arbitrary state and initially the network state is partly determined by the starting state Jaeger (2005). The reservoir needs time to become determined by the input and so training should not start from the beginning of the input. This means that some data is needed before encountering the first example of the non-base class.

The two resultant sub-files are shown in figures 4.16 and 4.17. The long pre-activity sequence can be seen in the first sub-file and the somewhat shorter post-activity sequence when the participant was taking off the sensor can be seen in the second sub-file.

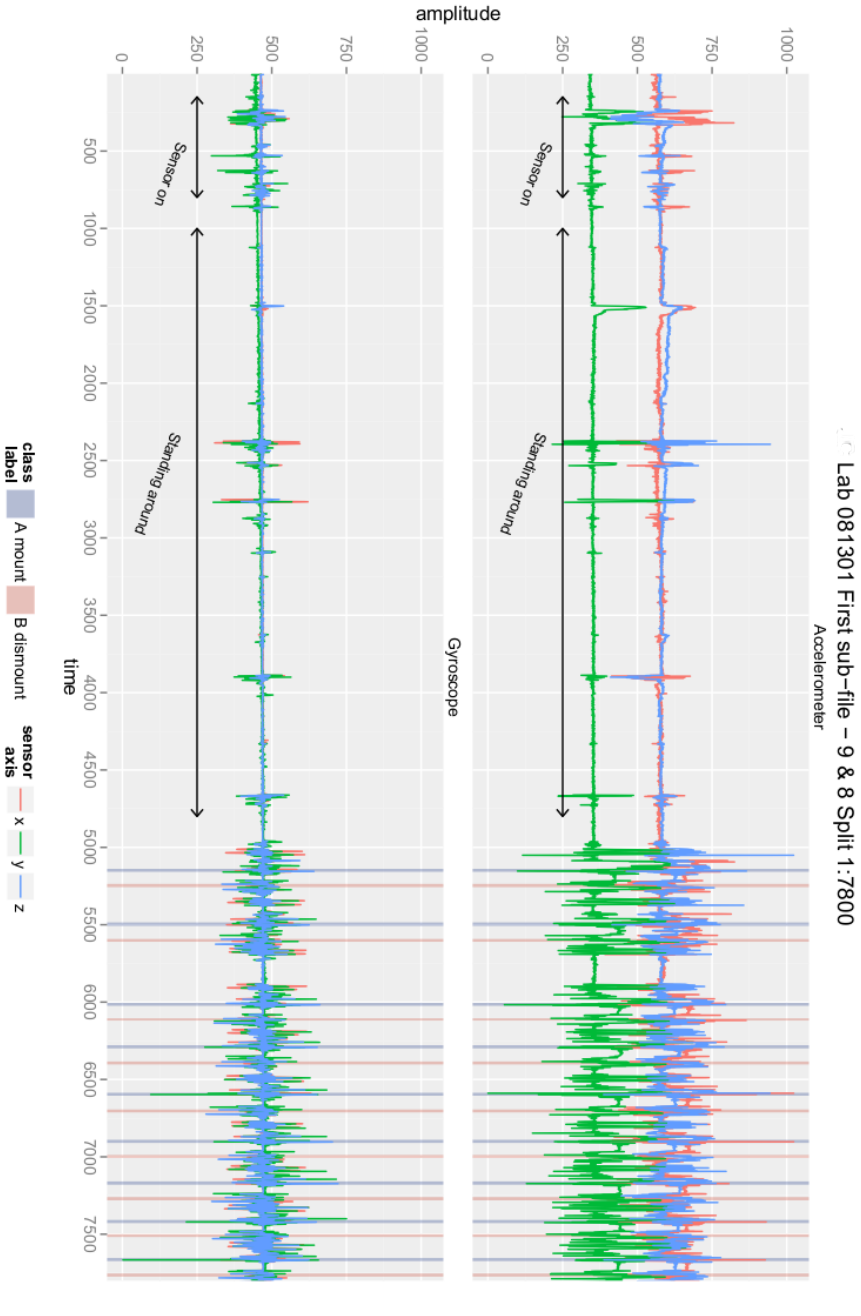


Figure 4.16: First LALab Raw data sub-file 1:7800

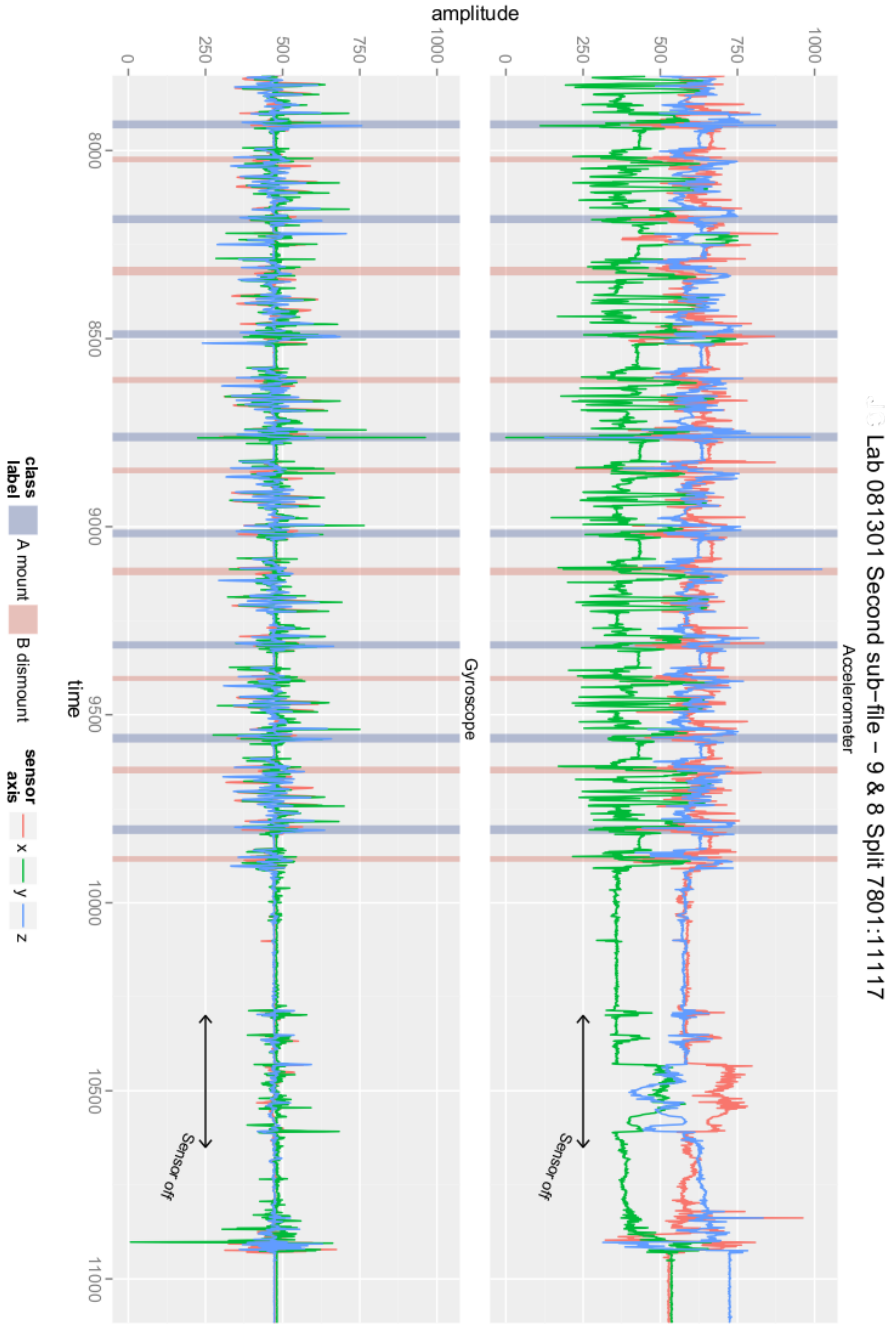


Figure 4.17: Second LALab Raw data sub-file 7801:11117

With LBLab (40 mount/dismount pairs) a similar process was followed and a decision was made to split this file into four files, each containing 10 mount/dismount pairs. This split meant it was possible to have separate files for tuning (train and test – using LBLab subsets) and then could concatenate one sub-file each from LALab and LBLab to create training and test sets for classification that contained balanced numbers of mounts and dismounts from each participant.

Chapter 5

INITIAL DESIGN CYCLE

This chapter describes the initial work that was done to demonstrate that RC techniques were a feasible technique for classifying unsegmented activity data and then the subsequent work to expand the use and evaluation of RC techniques for classifying unsegmented activity data captured from scripted activities within a laboratory environment.

5.1 INTRODUCTION

The design cycle characteristic of Design Science research methods has been separated into three phases for this work. Each phase having its own set of goals and measures of success. The first phase is a Proof-of-concept phase where the goal is to demonstrate that the concept of a punctual activity classifier based on RC techniques and using spatio-temporal data is feasible. The first step within phase 1 involves using generated, synthetic spatio-temporal data as input into the classifier and this work culminates in the publication of Schliebs and Hunt (2012). Once there is confidence that RC techniques might have a useful application classifying synthetic spatio–temporal data then the next step is to demonstrate this using more realistic data. This second step culminates in the publication of Schliebs et al. (2013). Both of these steps are described in Section 5.2. During this proof–of–concept phase a LSM is used to classify the activities of interest.

The second phase involves exploring the data from scripted, laboratory based activities and creating a tool-set and a Researcher's Workbench for RC punctual activity classifiers. This work is described in Section 5.4 and generated Hunt et al. (2014). At the start of this second phase the RC classifier model is changed from a LSM model to an ESN model for ease of use and for performance improvements during the search of the parameter space associated with each RC technique.

The third phase involves using the Researcher Workbench and RC classifier tool-set to create a RC based classifier that is capable of classifying real-life activities of interest. This work has its own chapter, namely Chapter 6 and is separately reported in Hunt and Parry (2015).

Within each phase, the particular design goals for that phase is discussed, along with any activity descriptions that are local to that phase.

5.2 PROOF OF CONCEPT (SYNTHETIC DATA)

This phase explores the feasibility of using RC techniques to classify punctual activity. The literature review in Chapter 2 demonstrated that RC techniques had not previously been used to classify human activity. As a result of that, it was not clear that RC techniques **could** be used to classify human activity. The initial problem, was then, to demonstrate that RC techniques could, in fact, be used for this purpose.

The motivation for this part of the research was to provide a proof-of-concept investigation of the applicability of a form of RC as the core component of a punctual activity classifier of un-windowed pseudo and realistic inertial data. Maass et al. (2002) was chosen as the basis for the RC model within this phase. Stage one of this phase, using synthetic data was reported in Schliebs and Hunt (2012), while stage two, using realistic data was reported in Schliebs et al. (2013).

LSM technology has been shown to be effective as a classifier in other areas such as continuous speech recognition (Schrauwen, D'Haene, Verstraeten, & Campenhout, 2008). Sensor data streams for continuous speech have some aspects in common with inertial data streams (spatio-temporal, un-windowed continuous data stream, variable length "tokens", digitally encoded analogue data) and so this gave us enough confidence to investigate this technology. In addition, the Technical Consultant, Dr Stefan Schliebs, had access to LSM code written in Python and this provided a convenient starting point.

The next sub-sections describes the design goals and measures, the LSM model, a possible classifier framework, the synthetic data, how the data was generated, the two encoding techniques chosen and reports the results along with a discussion.

5.2.1 *Design Goal - LSM with Synthetic Data*

The Design Goal for this part of this proof-of-concept phase was to create a LSM RC model that is capable of successfully classifying synthetic, un-windowed, spatio-temporal data sets into base, A & B classes.

5.2.2 *Measure of Success - LSM with Synthetic Data*

The measure of success was to consider a class as being correctly classified if the model output was greater than 60% of the difference between the base class signal and the expected class signal for at least 50% of the class width.

5.2.3 *Model description - LSM*

A LSM has two major components, a reservoir or "liquid" in the form of a RSNN (Gerstner & Kistler, 2002) and a trainable readout function.

The liquid is stimulated by spatio-temporal input signals causing neural activity in the Spiking Neural Network (SNN) that is further propagated through the network due to its recurrent topology. As a result, a snapshot of the neural activity in the

reservoir contains information about the current and past inputs to the system. The function of the liquid is to accumulate the temporal and spatial information of all input signals into a single high-dimensional intermediate state in order to enhance the separability between network inputs. The readout function is then trained to transform this intermediate state into a desired system output.

5.2.4 *Reservoir Description - LSM*

For the reservoir, we employ the LIF neuron which is arguably one of the best known models for simulating SNN. This neural model is based on the idea of an electrical circuit containing a capacitor with capacitance C and a resistor with a resistance R, where both C and R are assumed to be constant. The dynamics of a neuron i are then described by the following differential equations:

$$\tau_m \frac{\partial V_i}{\partial t} = -V_i(t) + R\, I_i^{\text{syn}}(t) \tag{5.1}$$

$$\tau_s \frac{\partial I_i^{\text{syn}}}{\partial t} = -I_i^{\text{syn}}(t) \tag{5.2}$$

The constant $\tau_m = RC$ is called the membrane time constant of the neuron. Whenever the membrane potential V_i crosses a threshold ϑ_i from below, the neuron fires a spike and its potential is reset to a reset potential V_r. We use an exponential synaptic current I_i^{syn} for a neuron i modelled by Eq. 5.2 with τ_s being a synaptic time constant.

In a similar manner to Schliebs, Fiasché, and Kasabov (2012), we define a dynamic firing threshold as a separate differential equation:

$$\tau_\vartheta \frac{\partial \vartheta_i}{\partial t} = \theta - \vartheta_i(t) \tag{5.3}$$

where θ is the minimum firing threshold of the neuron and τ_ϑ is the time constant for the dynamic threshold. Whenever the neuron i emits a spike, its threshold ϑ_i is increased by a constant $\Delta\theta_i$, *i.e.* $\vartheta_i \leftarrow \vartheta_i + \Delta\theta_i$. A dynamic synapse model based on the STP proposed by Markram et al. (1998) is used to exchange information between connected neurons.

We construct a reservoir having a Small-world inter-connectivity (SWI) pattern as described in Maass et al. (2002). A recurrent SNN is generated by aligning 500 neurons in a three-dimensional grid of size $10 \times 10 \times 5$. In this grid, two neurons A and B are connected with a connection probability

$$P(A, B) = C \times e^{\frac{-d(A,B)}{\lambda^2}} \tag{5.4}$$

where $d(A, B)$ denotes the Euclidean distance between two neurons and λ corresponds to the density of connections which was set to $\lambda = 3$ in the simulations. Parameter C depends on the type of the neurons. We discriminate into excitatory (ex) and inhibitory (inh) neural types resulting in the following parameters for C: $C_{ex-ex} = 0.3$, $C_{ex-inh} = 0.2$, $C_{inh-ex} = 0.4$ and $C_{inh-inh} = 0.1$. The network contained 70% excitatory and 30% inhibitory neurons.

5.2.5 *LSM Framework*

A possible framework for activity classification using LSM is illustrated in Figure 5.1.

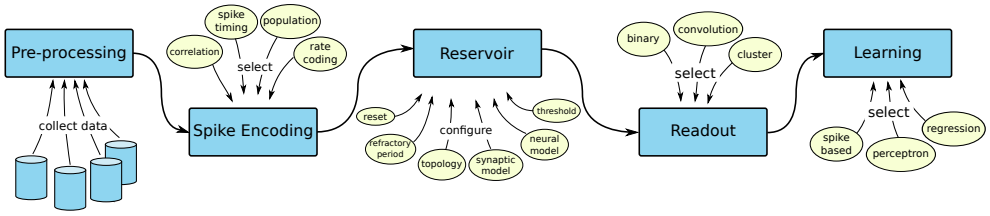


Figure 5.1: Possible framework to classify activities of interest

In the first step, the collected raw sensory time series data is pre-processed. Typical data cleaning procedures would be normalization, feature selection and outlier detection. In cyclic activity classification Fourier or wavelet transformations may also be considered as part of pre-processing. The synthetic data is generated in a state that does not require pre-processing.

The cleaned signal is then encoded into an input compatible with the reservoir. Numerous encoding algorithms have been proposed in literature and we consider two common techniques as part of this initial research. The encoded input is then fed into the reservoir where it results in a temporal change of neural activity in the RSNN which is read out periodically. A machine learning algorithm then learns the mapping from the extracted readouts to a desired class label.

Several design and configuration choices have to be made for each component of the framework. The ellipses in Figure 5.1 show some possible options for these decisions. Each framework component is explained in greater detail and some of the design decisions for this design iteration are indicated in the following sections.

5.2.6 *Synthetic Data Description and Preprocessing*

We decided to generate a synthetic multi-sensor time series data set so that we would have control over aspects such as the number of sensor signals, problem difficulty and signal-to-noise ratio. Furthermore, this data set provides an ideal solution, since the exact differentiation between signal and noise is known a priori, and it allows the analysis of the proposed technique for a large variety of testing scenarios.

The constructed N synthetic time series are generated as a superposition of C sine waves. Signal $s_n(t)$ is described as

$$s_n(t) = \sum_{i=1}^{C} a_n^{(i)} \sin(\omega_n^{(i)} t + \phi_n^{(i)}) \tag{5.5}$$

where $C \in \mathbb{N}$ is the number of superimposed functions and the parameters $a_n^{(i)} \in \mathbb{R}$, $\omega_n^{(i)} \in \mathbb{R}$, $\phi_n^{(i)} \in \mathbb{R}$ represent the amplitudes, frequencies and phases of the individual sine functions, respectively. For the study, we chose $C = 5$ superimposed sine waves that were parametrized with a set of random amplitudes, frequencies and phases. The

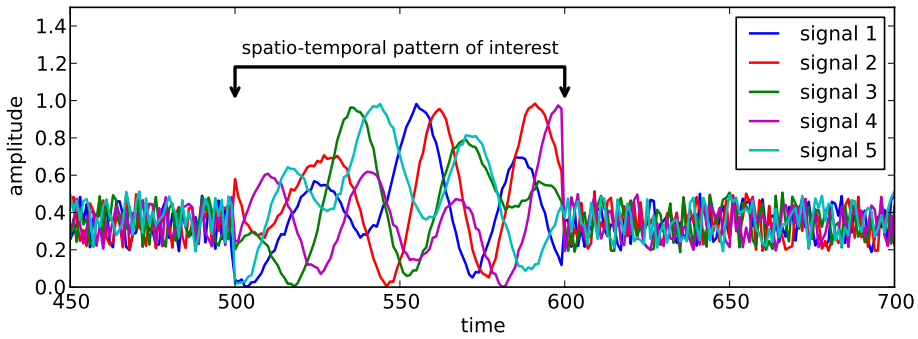


Figure 5.2: Synthetic data investigated as a proof of concept

$N = 5$ time series $s_n(t)$ were "pasted" into a uniformly distributed noise sequence at various locations and a small Gaussian noise ($\sigma = 0.01$) was added to the resulting signal. Finally, we normalized the signal to be between $[0, 1]$. Figure 5.2 depicts part of the generated time series.

5.2.7 *Synthetic Data Encoding*

The time series data obtained from the sensors are presented to the reservoir in the form of an ordered sequence of real-valued data vectors. In order to compute an input compatible with the SNN, each real value of a data vector is transformed into a spike train using a spike encoding technique. We explore two different encoding techniques in this work, namely Ben's Spike Algorithm (BSA) (Schrauwen & Van Campenhout, 2003) and a Population encoding technique.

BSA assumes the analogue signal to be a convolution of a spike train. The algorithm attempts to estimate the corresponding spike train responsible for this convolution by reversing the convolution process. The technique has a threshold parameter which we set to 0.955 and we use the discrete linear filter presented in de Garis, Nawa, Hough, and Korkin (1999). More detailed explanations of this encoding along with pseudo code can be found in Schrauwen and Van Campenhout (2003).

In contrast to BSA, the Population encoding uses more than one input neuron to encode a single time series. The idea is to distribute a single input to multiple neurons, each of them being sensitive to a different range of values. This implementation is based on arrays of receptive fields with overlapping sensitivity profiles as described in Bohte, Kok, and Poutré (2002); Schliebs, Defoin-Platel, and Kasabov (2009). The reader is referred to the mentioned references for further details and examples of this encoding algorithm. As a result of the encoding, input neurons emit spikes at predefined times according to the presented data vectors.

5.2.8 *Readout and Learning - LSM with Synthetic Data*

In this part of the study, we use the typical analogue readout function in which every spike is convolved by a kernel function that transforms the spike train of each neuron

in the reservoir into a continuous analogue signal. We use an exponential kernel with a time constant of $\tau = 50$ms. The convolved spike trains are then sampled using a time step of 10ms resulting in 500 time series – one for each neuron in the reservoir. In these series, the data points at time t represent the readout for the presented input sample. A very similar readout was used in many other studies, *e.g.* in Schrauwen et al. (2008) for a speech recognition problem.

Readouts were labelled according to their readout time t. If the readout occurred at the time when a sensor signal of interest (*e.g.* mounting/dismounting the horse) was fed into the reservoir, then the corresponding readout is labelled as a class-A or class-B sample. Consequently, a readout belongs to the base class, if it was obtained during the presentation of a noise part of the input signal.

The final step of the LSM framework consists of a mapping from a readout sample to a class label. The general approach is to employ a machine learning algorithm to learn the correct mapping from the readout data. In fact, since the readout samples are expected to be linearly separable with regard to their class label Maass et al. (2002), a comparably simple learning technique can be applied for this task. From the labelled readouts, we obtained a linear regression model mapping a reservoir readout sample to the corresponding class label, *i.e.* either 0 (base class) or 2 (class A) or -2 (class B).

5.2.9 *Results - LSM with Synthetic Data*

Figure 5.3 shows the outputs obtained from each of the individual processing steps of the LSM framework. Sub-figure A shows the input data, sub-figure B shows the effect of the input on the BSA encoded input neurons, sub-figure C shows the effects of the same input on the Population encoded input neurons, sub-figure D shows the response from the output neurons when using BSA encoding, sub-figure E shows the internal liquid state when using BSA encoding, sub-figure F shows the response from the output neurons when using Population encoding, sub-figure G shows the internal liquid state when using Population encoding and sub-figure H shows a raster plot of the results of the weighted outputs of both the BSA encoded and Population encoded data after linear regression plus the training signal.

A set of five synthetic time series was generated over a time window of 4200ms which included ten occurrences of two alternating temporal patterns, *cf.* Figure 5.3A. The encoded spike trains derived from the given time series are depicted in Figure 5.3B (BSA encoding) and 5.3C (Population encoding), respectively. The figures show a raster plot of the neural activity of the input neurons over time. A point in these plots indicates a spike fired by a particular neuron at a given time.

The obtained spike trains were then fed into a reservoir resulting in characteristic response patterns for each encoding type, *cf.* Figure 5.3D and 5.3F. The small number of input neurons employed by the BSA encoding results in a rather sparse reservoir response. A repeating pattern of neural activity was observed, however, the relevance for the detection of the patterns of interest was less obvious. The response of the Population-encoded signal was denser and the impact of the non-noise signal on the reservoir was obvious from the raster plot.

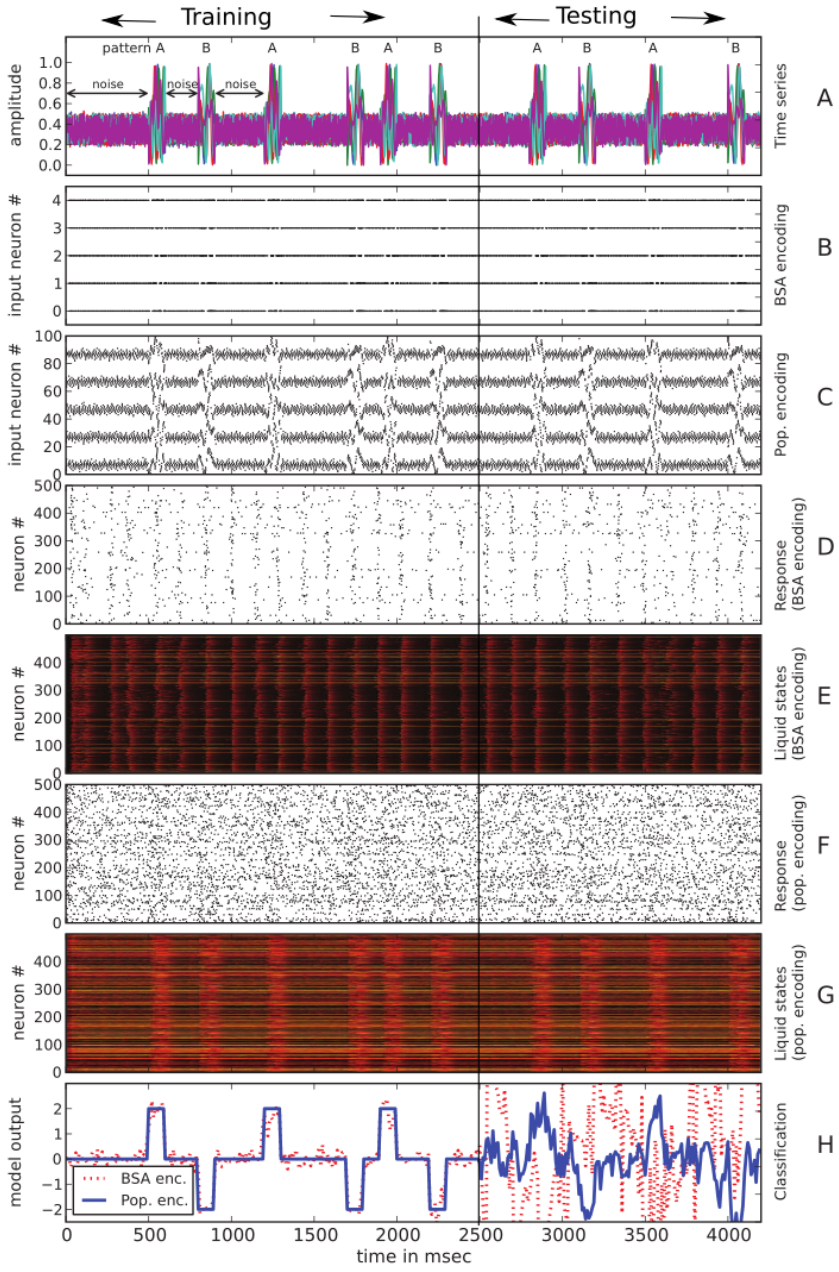


Figure 5.3: Results of the continuous classification of the synthetic data. See text for detailed explanation of the figure.

The reservoir was continuously read out every 10ms of the network simulation using the technique described in subsection 5.2.8 on page 98. Figures 5.3E and 5.3G show the readouts over time for BSA and Population-encoded reservoir inputs, respectively. The colour in these plots indicates the value of the readout obtained from a certain neuron; the brighter the colour, the larger the readout value. The bright horizontal lines in this plot indicate the reservoir neurons that are directly stimulated from the encoded spike trains of the input neurons. The Population-encoded stimulus

causes a very characteristic readout pattern in which the non-noise signals are clearly detectable.

The learning and classification step of the LSM framework is presented in the last plot of Figure 5.3H. We used a linear regression model that was trained with the first 2500ms of readout data and then tested on the remaining 1700ms of the simulation. A class-A (class-B) sample was considered as correctly classified, if the model output was larger (smaller) than a threshold of 1.2 (-1.2) for at least 50% of the class width. 1.2 being 60% of the output range for each non-null class.

5.2.10 *Discussion - LSM with Synthetic Data*

As already suspected from the reservoir responses of Figure 5.3D, the readouts obtained using BSA encoding of the input signal appear difficult to map to the correct class label, contrast the dotted (red) curve in Figure 5.3G with the input data. While the training data was learned well enough, the testing accuracy dropped to a random classification (around 24.7%). We assume an unsuitable configuration of the reservoir as the reason for this low performance.

On the other hand, the model responses for the readouts obtained using Population encoding showed much more promising results, compare the solid (blue) curve in Figure 5.3G with the input data. The model reported perfect classification on the training data and a satisfying classification accuracy on the testing data (93.5%). Errors usually occurred immediately after the onset of a signal when the reservoir had not yet accumulated sufficient information about the pattern. Considering the short training period, this result was very encouraging.

These results were promising enough to give confidence that a LSM utilising Population encoding may provide a suitable technique to detect spatio-temporal patterns in serial data such as realistic inertial data.

5.2.11 *How the design iterated*

The next step involved testing this technique on realistic inertial data from the laboratory data, rather than synthetic data.

5.3 PROOF OF CONCEPT (REALISTIC DATA)

This design iteration employs a LSM to classify real(istic) inertial sensor data collected from a single horse rider who was following a scripted set of actions in a laboratory situation. While the data collected is real data, in the sense that it is the actual data collected from the sensor during the data capture session, it is called realistic data rather than real data as the activities being monitored are scripted and conducted within a controlled laboratory environment and so the activities are not real-world activities. This work builds on the stage one work which used synthetic data. The following sections describe the design goals and measures for this part of the work, the realistic data, how the data was captured and pre-processed, the encoding technique chosen and reports the results with a discussion.

5.3.1 *Design Goal - LSM with Realistic Inertial Data*

The design goal for this iteration was to confirm the proof of concept applicability of LSM for the chosen problem domain by demonstrating a working LSM classifier using realistic data captured from an inertial sensor that was worn during real, scripted activities.

5.3.2 *Measure of Success - LSM with Realistic Inertial Data*

The measure of success was to consider a class as being correctly classified if the model output was greater than 60% of the difference between the base class signal and the expected class signal for at least 50% of the class width.

5.3.3 *Framework Description - LSM with Realistic Inertial Data*

This work employs the same general framework as previously described in subsection 5.2.5 and illustrated in Figure 5.1 on page 97. In addition, the LSM model and Reservoir are as described in subsections 5.2.3 and 5.2.4 on pages 95 and 96.

The initiative recently proposed in Nordlie, Gewaltig, and Plesser (2009) that promotes reproducible descriptions of neural network models and experiments is followed when describing this work. The initiative suggests the use of specifically formatted tables explaining neural and synaptic models along with their parametrization. The set-up outlined in Table 5.1 is used for this design iteration. Within the table the following acronyms: LIF; SWI; and STP have been used.

A reservoir having a SWI pattern as described in Maass et al. (2002) is constructed. A recurrent SNN is generated by aligning 500 neurons in a three-dimensional grid of size $10 \times 10 \times 5$. In this grid, two neurons A and B are connected with a connection probability

$$P(A, B) = C \times e^{\frac{-d(A,B)}{\lambda^2}} \tag{5.6}$$

Table 5.1: Description of LSM setup.

Model Summary	
Neural model	LIF with dynamic firing threshold Schliebs et al. (2012)
Synaptic model	Exponential synaptic currents
Input	Population encoded, normalized, real-valued data
Connectivity	SWI between reservoir neurons
Neural Model	
Type	LIF neuron
Description	Dynamics of membrane potential $V(t)$: • Spike times: $t^{(f)} : V(t^{(f)}) = \vartheta(t^{(f)})$ • Sub-threshold dynamics: $\tau_m \frac{\mathrm{d}V}{\mathrm{d}t} = -V(t) + R\, I^{\mathrm{syn}}(t)$ $\tau_\vartheta \frac{\mathrm{d}\vartheta}{\mathrm{d}t} = \theta^{\min} - \vartheta(t)$ • Reset and refractoriness $\forall f : t \in (t^{(f)}, t^{(f)} + \tau_{\mathrm{ref}})$: $\vartheta(t) \leftarrow \vartheta(t) + \Delta\theta$ $V(t) \leftarrow V_r$ • Exact integration with temporal resolution dt
Parameters	Membrane time constant $\tau_m = 30$ms Membrane resistance $R = 1\mathrm{M}\Omega$ Threshold: $\theta^{\min} = 0.5$mV, $\Delta\theta = 5$mV, $\tau_\vartheta = 50$ms Refractory period $\tau_{\mathrm{ref}} = 1$ms, reset potential $V_r = 0$mV Time resolution $dt = 0.1$ms, simulation time $T = 6500$ms
Synaptic Model	
Type	Current synapses with exponential post-synaptic currents
Description	Synapse modelled using STP Markram et al. (1998): $\tau_s \frac{\mathrm{d}I^{\mathrm{syn}}}{\mathrm{d}t} = -I^{\mathrm{syn}}(t)$ $\tau_D \frac{\mathrm{d}x_i}{\mathrm{d}t} = 1 - x_i$ $\tau_F \frac{\mathrm{d}u_i}{\mathrm{d}t} = U_i - u_i$ Pre-synaptic spike from neuron j to neuron i triggers: $I_i^{\mathrm{syn}} \leftarrow I_i^{\mathrm{syn}} + w_{ji}\, x_i\, u_i$ $u_i \leftarrow u_i + U_i(1 - u_i)$ $x_i \leftarrow x_i(1 - u_i)$
Parameters	Synaptic weight $w \in \mathbb{R}$, uniformly initialized in $[-50, 50]$nA Synaptic time constant $\tau_s = 5$ms, Utilization $U = 0.1$, Depression $\tau_D = 500$ms, Facilitation $\tau_F = 200$ms

where $d(A, B)$ denotes the Euclidean distance between two neurons and λ corresponds to the density of connections which was set to $\lambda = 3$ in the simulations. Parameter C depends on the type of the neurons, excitatory (ex) and inhibitory (inh) neural types, resulting in the following parameters for C: $C_{ex-ex} = 0.3$, $C_{ex-inh} = 0.2$, $C_{inh-ex} = 0.4$ and $C_{inh-inh} = 0.1$. The network contained 70% excitatory and 30% inhibitory neurons that were chosen randomly.

5.3.4 *Sensor Description*

As described in Section 4.6, the SparkFun 6DoF inertial sensor contains a Freescale MMA7260Q triple-axis accelerometer, two InvenSense IDG300 500^o per second gyroscopes and both a Honeywell HMC1052L and a HMC1051Z magnetic sensor. The sensor outputs readings from a 12 bit analogue to digital converter that gives a reading range between 0 and 1023.

Sensor readings were sampled at 10Hz and broadcast via Bluetooth to an on-body receiver for logging and later analysis. The accelerometer in this sensor has a settable scale and for this session it was set to record $\pm$2G's.

5.3.5 *Activity Definitions*

The formal definitions for *stirrup mount* and *dismount*, as described in section 2.2 on page 11 are used within this design iteration. In addition, it was decided to add a buffer of 10 samples (1 second at 10Hz) on either side of the start and end point of each activity. This was done to encapsulate some small level of pre-mount and post-mount activity. The reason for adding the buffer to the start was that the author expected the LSM to take some time to ramp up and to correctly classify the activities, based on experience with the synthetic data classification. By including the start buffer, it was hoped that this would improve the correct classification of the activity closer to its start point. The post-activity buffer was added in the expectation that this might enhance the classification success by including a short phase of 'mounted' data which may have been consistent across riders.

5.3.6 *Realistic Data Description and Preprocessing*

The data is as described in Chapter 4.6 and is one of the two laboratory data-sets (LALab) with data from a single participant who followed the given activity script (including mounting and dismounting) 17 times. The SparkFun (SparkFun Electronics Inc, 2008a) inertial sensor used is as described in Chapter 4.6.

Data collection was done as part of a Masters project (Hunt, 2009) and all data was collected within a laboratory on the KTH Kista campus in Sweden. The session was videoed so that activities could be manually classified by the research team.

The participant mounted and dismounted the wooden horse following a similar technique to that previously illustrated in Figure 2.1 on page 13, except that mounting and dismounting were simpler and safer (indoors) as the wooden horse was stable

enough to not move during the data capture session. An example of the laboratory mounts and dismounts is shown in Figures 5.13 and 5.14 on pages 128 and 129.

During this session the rider wore the sensor on the right wrist using a simple stretchable Velcro bandage for attachment. The "horse" used during the laboratory sessions was a built-for-purpose wooden framed horse of approximately 16 hands in height (163cm at the "shoulder"), draped with a standard European riding saddle and stirrups (see Figure 5.4). During this particular laboratory session the rider mounted and dismounted 17 times.

Figure 5.4: A laboratory participant stands ready to start mounting the wooden horse.

The output from all three inertial sensors (magnetometer, accelerometer and gyroscope) were used for this part of the work. Figure 5.5 depicts part of the recorded time series. The three upper panels show the 3-dimensional recordings from the magnetometer, accelerometer and gyroscope, respectively. The bottom panel shows the activity undertaken during recording (mounting/dismounting). The figure shows two mounts and two dismounts.

The participant was asked to mount and dismount as closely as possible to her normal technique and apart from the requested pauses was not asked to keep to any particular time schedule. The duration of each mount and dismount is, nevertheless, reasonably consistent with a gradual shortening of duration as the participant gets used to and more proficient at the script.

The first mount takes 11 seconds while the last mount takes 6.5 seconds with progressive shortening in between. The first dismount takes 9 seconds and the last dismount takes 6.7 seconds again with progressive shortening. There is also a small shortening of the interval between each mount/dismount pair with the first interval being 11.3 seconds and the last being 7.2 seconds. However there is also a longer interval after the first two pairs.

The interval between each mount and dismount (when the participant is sitting on the horse) is much more consistent (probably as a result of the script). The first and second mount to dismount intervals are both 4 seconds and the last mount to dismount interval is 3.4 seconds.

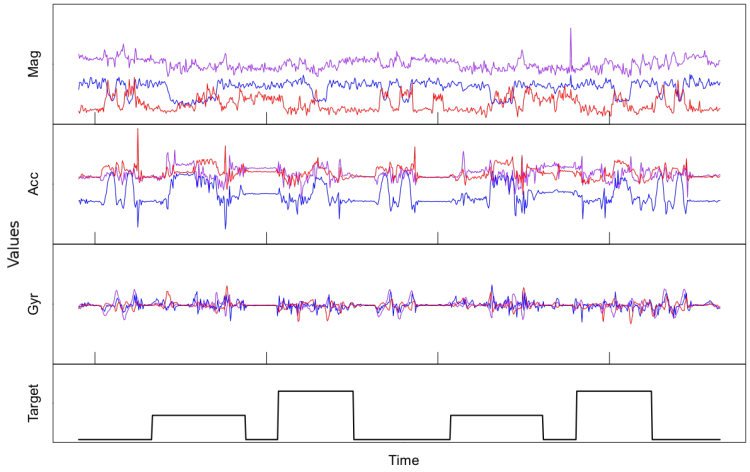


Figure 5.5: Subset of sensor data collected from a participant during a laboratory session.

The periods prior to the mount/dismount series records the sensor on a bench after being turned on, being fitted to the participant's wrist and then the participant waiting around while the researcher checked and adjusted equipment such as the video camera. The period after the series is essentially the participant taking off the sensor and it being placed back on a bench.

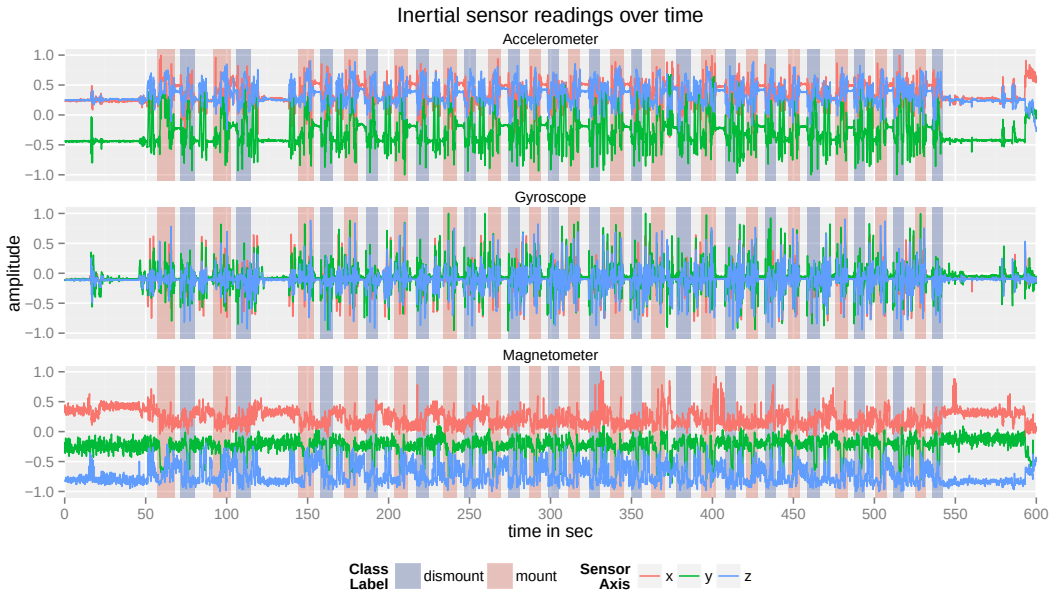


Figure 5.6: LALab Input Data Set

Figure 5.6 depicts the recorded time series after pre-processing and manual labelling. The three panels each show the 3-dimensional recordings from the accelerometer, gyroscope and magnetometer respectively. The alternating background

stripes show the activity undertaken during recording (mounting, dismounting and null class). The figure depicts all 17 mounts and dismounts and the figure has been truncated to the right to help highlight the mount and dismount data.

From figure 5.6, careful observation shows that there is a slight drift upwards over time with the Gyroscope data. This drift is relatively common with lower priced Gyroscopes and this tendency has been ignored in the data analysis. In addition, the y axis (*green*) of the Magnetometer data also drifts upwards over time, while the x and z axis do not show the same level of drift. The presence of the drift on only a single axis is possibly due to the sensor moving slightly on the participant's wrist as they move about. In addition, some level of drift is common with consumer level Magnetometers such as this one but normally the drift would be present on all axis.

The level of drift in the raw magnetometer data represents 3.5% of the possible range, over 17 minutes but once the Magnetometer data is cleaned and normalised the drift is magnified and represents a 32.7% difference in range over the full 17 minutes. Cleaning and normalisation has expanded the central part of the signal and emphasised the drift. However, most of the drift occurs before the first mount and after the last dismount. Between the first mount and last dismount the drift in the cleaned and normalised signal is only slightly more than 1.69% of the full range and so for these iterations this Magnetometer drift has also been ignored.

5.3.7 *Input Data Editing & Error Checking*

The raw data files are hand edited to remove extraneous set up data and commands that were logged before and after the main data file. Column labels are inserted as the first line of the cleaned file. An error checking routine is then run against the file to check for missing and out of range data. No out of range, invalid or missing data was found in this file.

5.3.8 *Data Synchronisation & Labelling*

Synchronisation was done following the process outlined in Figure 4.1 and described in Section 4.6.3, see page 79. Once synchronisation is complete the video is used to find the frame where the occurrence of each activity starts and finishes. The video frame numbers are then translated into an applicable sample numbers from the sensor stream, based on the synchronisation point, and class labels are added to the sensor data based on the activity definitions.

5.3.9 *Realistic Data Encoding*

The time series data obtained from the sensors are presented to the reservoir in the form of an ordered sequence of real-valued data vectors. In order to compute an input compatible with the SNN, each real value of a data vector is transformed into a spike train using a spike encoding. Based on the earlier work with synthetic data

we decided to use Population encoding only, as this was the only technique that gave satisfying results.

Population encoding uses more than one input neuron to encode a single time series. The idea is to distribute a single input to multiple neurons, each of them being sensitive to a different range of real values. This implementation is based on arrays of receptive fields with overlapping sensitivity profiles as described in Bohte et al. (2002); Schliebs et al. (2009). The reader is referred to the mentioned references for further details and examples of this encoding algorithm. As a result of the encoding, input neurons emit spikes at predefined times according to the presented data vectors.

5.3.10 *Readout and Learning - LSM with Realistic Data*

In this part of the study, we use the typical analogue readout function in which every spike is convolved by a kernel function that transforms the spike train of each neuron in the reservoir into a continuous analogue signal.

We use an exponential kernel with a time constant of $\tau = 50$ms. The convolved spike trains are then sampled using a time step of 10ms resulting in 500 time series – one for each neuron in the reservoir. In these series, the data points at time t represent the readout for the presented input sample. A very similar readout was used in many other studies, *e.g.* in Schrauwen et al. (2008) for a speech recognition problem.

Readouts were labelled according to their readout time t. If the readout occurred at the time when a sensor signal of interest (*e.g.* mounting/dismounting the horse) was fed into the reservoir, then the corresponding readout is labelled accordingly. Consequently, a readout belongs to class 0 (base class), if it was obtained during the presentation of a part of the input signal that is of no particular interest.

The final step of the LSM framework consists of a mapping from a readout sample to a class label. The general approach is to employ a machine learning algorithm to learn the correct mapping from the readout data. In fact, since the readout samples are expected to be linearly separable with regard to their class label Maass et al. (2002), a comparably simple learning technique can be applied for this task. From the labelled readouts, we obtained a ridge regression model for each activity of interest mapping a reservoir readout sample to the corresponding class label. Ridge regression is essentially a regularized linear regression that has been reported to counteract model over-fitting. We tried different values for the regularization parameter α and used $\alpha = 10$ for this design iteration.

5.3.11 *Results - LSM with Realistic Data*

Figure 5.7 shows the outputs obtained from each of the individual processing steps of the LSM framework. Sub-figure A shows the input data, sub-figure B shows the effect of the input on the Population encoded input neurons, sub-figure C shows the response from the output neurons when using Population encoding, sub-figure D shows the internal liquid state when using Population encoding and sub-figure E shows the results of the weighted outputs of the Population encoded data after linear regression plus the training signal for both Mounts and Dismounts.

The data consists of a set of nine time series over a time window of 6500ms of simulation time which included 17 occurrences of two alternating mounting and dismounting patterns, *cf.* Figure 5.7A. The encoded spike trains (Population encoding) derived from the time series are depicted in 5.7B. The figures show a raster plot of the neural activity of the input neurons over time. A point in these plots indicates a spike fired by a particular neuron at a given time.

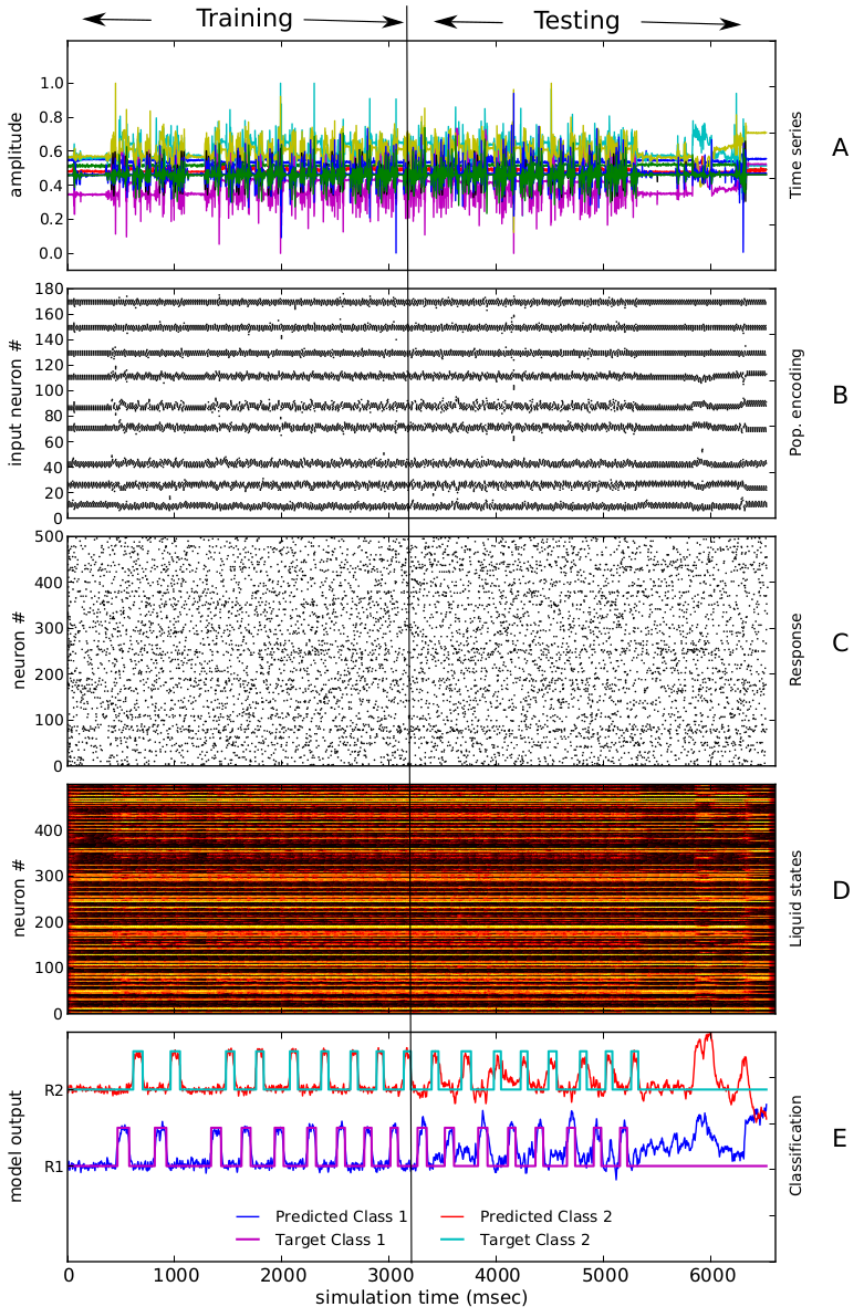


Figure 5.7: Results obtained from the LSM using realistic data as input.
See text for detailed explanations on the figure.

The obtained spike trains were then fed into a reservoir resulting in characteristic response patterns of the reservoir neurons, *cf.* Figure 5.7C. The reservoir is continuously read out every 10ms of the time simulation using the technique described in section 5.3.10. Figure 5.7D shows the readouts over time for the Population-encoded reservoir inputs. The colour in these plots indicates the value of the readout obtained from a certain neuron; the brighter the colour, the larger the readout value. The bright horizontal lines in this plot indicate the reservoir neurons that are directly stimulated from the encoded spike trains of the input neurons. The stimulus causes a characteristic readout pattern in which the mount and dismount signals are detectable.

The learning and classification step of the LSM framework is presented in the last plot of Figure 5.7E. Within this figure, graph R1 (class 1) represents Mounts and graph R2 (class 2) represents the Dismounts. Within R1 the target signal (Mounts) is shown in dark red with the output (predicted) signal shown in dark blue and overprinted. Within R2 the target signal (Dismounts) is shown in light blue and the output (predicted) signal in light red.

We used two ridge regression models, one for learning the mounting class and the other for learning the dismounting class. Both ridge regression models were trained on the first 3250ms of readout data and then tested on the entire set of time series. For the testing, we obtain the output of both regression models and choose the one reporting the larger output in order to decide the class label (winner-takes-all strategy).

Contrasting the two regression models associated with R1 and R2 it can be seen from the figure that, in this case, the models was able to clearly differentiate Mounts from Dismounts in all cases where actual Mounts or Dismounts occurred. At around 3800 the predicted class 1 signal (Mount) is showing some additional amplitude during a dismount activity but the class 1 amplitude is clearly much lower than the class 2 amplitude and so this dismount is correctly classified. At around 4000 the class 2 (dismount) amplitude is raised during a mount activity but again the class 1 amplitude is much higher than the class 2 amplitude during this sequence and so, again, the activity is correctly classified. However, at around 5900 both the class 1 and class 2 signal amplitudes increase during a period when nether a mount nor a dismount is occurring and so both signals produce a false positive although as the class 2 amplitude is higher only a false positive dismount would result. The opposite occurs at around 6300. In both cases the actually activity being performed by the subject is that they are taking off the sensor from their wrist.

A sample was considered as correctly classified, if the corresponding model output was larger than a threshold of 0.6 for mount/dismount samples. The model reported excellent classification on the training data (97.2%) and a satisfying classification accuracy on the testing data (85.1%). Errors usually occurred immediately after the onset of a signal when the reservoir had not yet accumulated sufficient information about the pattern. Considering the short training period, this result is encouraging. The classification accuracy decreases at the very end of the simulation. Here the horse rider actually removes the sensor from her wrist which is an activity not trained within the system resulting in highly variable model outputs including false positive classification.

5.3.12 *Discussion - LSM with Realistic Data*

This part of the research takes the initial ideas and framework that were applied to synthetic data and applies these ideas and framework to realistic data captured during a scripted activity session from a single participant (rider) under laboratory conditions. The LSM model that was produced met and exceeded the design goals, successfully classifying the two activities of interest although also producing false-positive results as well.

While these results are encouraging, it should be acknowledged that the data does not include any of a myriad of common activities normally encountered by horse riders and where one such activity was included (taking off the sensor device) this resulted in a false positive being detected for both classified activities. Countering this caution, none of these activities were included within the training sub-set of the data and so (perhaps) it is unreasonable to be too critical of the false-positive detections until more varied data is included within the training set.

In addition, the script that the participant followed while recording this data was very regular, with regular pauses and repetition of a small subset of activities with the same orientation (horse to rider) and within a confined geographic area. This regularity creates artificial artefacts within the data and, unfortunately, it is not possible to tell what artefacts of the data the LSM is learning and so it is not know how well this will translate into a real-world situation or even across multiple riders within a scripted, laboratory environment.

At this stage the author was reasonably confident that RC techniques in general and LSM based models in particular were capable of classifying spatio-temporal inertial sensor data. However, this data has not been tested with alternate, windowed and more traditional classifiers and so at this stage it is not know if RC techniques have advantages or disadvantages over more traditional techniques.

A LSM is a complex system with a large number of parameters needed to configure spike encoding, reservoir size and connectivity, readout and learning algorithms. A grid search was used to select feasible parameters for this design iteration and a typical run of the grid search, utilising LSM models with 500 neurons and repeating each train and test cycle three times to enable us to average the model response took between five and seven days of elapsed processing time on a MS Windows i7 PC. Such long runs were prone to problems in a networked, corporate type computing environment subject to enforced software updating and the occasional power outage. If a problem occurred during such a long run then a lot of time was lost re-running the model. A solution that did not need such long run times was highly desirable.

Despite this the results that were obtained were promising enough to convince the author that the concept of using a RC based classifier was feasible enough to progress on to simplifying the overall technique, decreasing run times for model testing and applying RC techniques to multiple participants and real world data. What was clear, though, was that LSM techniques in particular required considerable tuning of the largish numbers of parameters in order to produce reasonable results.

5.3.13 *How the design iterated*

The proof-of-concept work had provided enough confidence to continue down the track of using non-windowed RC techniques as a classifier for punctual activities but it was also clear that there was a need to try more complex (and perhaps larger) datasets, additional parameter tuning and result visualisation. The existing tools were complex to operate and required a lot of processing power to run.

A wider view was taken and the author looked at other possible tool-sets that would allow for similar things to be done but perhaps in a less complex manner, using tool-sets that were more generalised so that existing libraries could be used, where possible, for alternate techniques and where more easily parallelisable versions of routines or code could be found that would reduce the elapsed compute time so that the long compute times needed for optimising parameter sets could be made more manageable.

5.4 EXPLORATION PHASE (PHASE 2)

The aim of this second phase of the research was to explore the various forms of RC, parameter selection and input data pre-processing to obtain a workable set of tools to allow the goal driven research to progress at a quicker rate. A summary of the results from this phase of the work were reported in Hunt et al. (2014).

Phase one work has shown promise of a potentially reliable classification technique for punctual movements, however the ability to further explore this problem area has been constrained by a need to optimise the run-time parameters for the LSM. Early efforts were based around using a simple grid search to find useful parameter sets based on heuristic advice on likely good starting points. Some of these early grid searches took between five and seven days of continuous running on a single i7 MS Windows based PC using code developed by Schliebs in Python. A number of the datasets contain in excess of 36,000 samples of sensor readings and none of the datasets in the current collection contain less than 10,000 sensor reading samples. Training and testing a LSM on datasets of this size paired with numbers of neurons in (perhaps) the low thousands requires significant computing resources using then available techniques and equipment. While this was workable for one-off iterations once there was a useful set of LSM run-time parameters matched to the dataset, it was much less workable when trying to optimise the LSM run-time parameters.

There are a number of ways to either reduce the computing resources required or increased the available compute power, such as:

1. optimising and re-implementing the python code using C for key areas;
2. parallelising the code so that it could take advantage of multi-processor i7 and similar CPU's;
3. redesigning the code to use GPU's for key, compute intensive parts; running the grid searches on a networked Condor cluster or running the GPUised code on a local GPU cluster;

4. using alternate, potentially more computationally efficient RC techniques and changing the parameter search methodology to something a bit smarter than a simple grid search.

5.4.1 *Performance Improvement Problem Identification*

The long, multiple day, LSM parameter optimisation runs took too long to run and when run on a desktop computer in a corporatised (university) environment sometimes failed prior to completion as a result of power outages (cleaners or others turned off computers at night) or corporate enforced reboots after system and security updates. This meant that potentially a run might take even longer with re-starts with the worst recorded case taking over three weeks to run a single optimisation with three restarts. The extended time between having an idea about how something could be improved, developing the code to implement that idea and then demonstrating that idea in practise was proving impractical and was restricting any desire to try new ideas. The period of time between having the idea and demonstrating the idea needed to be reduced.

5.4.2 *Performance Improvement Motivation*

The motivation for this design iteration was to improve the available tool-set by speeding up the time between ideation and demonstration so that it became possible to have more ideation–demonstration cycles. This, in turn, would allow for more comprehensive demonstrations of the value of using RC models to classify punctual activities based on inertial spatio-temporal data.

As a first step, it was decided to try using ESN technology as an alternative to LSM technology to see if this would give similar error rates during testing while reducing compute requirements. Work such as Verstraeten et al. (2007) provides an excellent comparison of LSM and ESN techniques across different dataset types and they found that the use of spiking techniques (LSM) were generally more beneficial at equivalent neuron population sizes but did not report computational requirements. Our own experience suggested that ESN's, in general, required less computational resources but there was no imperative evidence to test if increasing the ESN neuron population size could produce equivalent error rates to LSM techniques with smaller neuron populations while retaining the potential computational advantages of ESN. A decision was made to implement ESN techniques in R (R Core Team, 2012) and re-run the initial LSM Realistic Data iteration (Schliebs et al., 2013) to see if this approach had any advantages. Schliebs wrote an R routine that implements ESN as defined by Lukoševičius and Jaeger (2009, pp 127-149).

5.4.3 *Performance Improvement Design Goals*

The goals of this first exploratory step was to ensure:

1. that the newly written ESN R code worked as planned by achieving comparable results vis-a-vis the Python based LSM code
2. that the partially parallelised version of the PSO search ran with considerably less elapsed time than the grid-based search, in an appropriate environment.

5.4.4 *Performance Improvement Measures of Success*

1. A classification rate similar to that achieved by the LSM model
2. An elapsed compute time for the parameter optimisation process that is considerably less than that for a grid search with a similar parameter range.

5.4.5 *Performance Improvement Description*

The first step in this phase was to repeat the LSM work with the LALab data using the newly written ESN routines in R. In addition, it was decided to try to speed up the model parameter optimisation searches by switching from a grid-based search methodology to an evolutionary algorithm, specifically a PSO. PSO technology has been recommended by researchers such as Lin, Ying, Chen, and Lee (2008) because of their ability to cover a wide search space relatively quickly.

R immediately proved to be a good choice as it has an existing library routine designed to implement PSO optimisation techniques (Bendtsen., 2012). This proved to be a simple call to a '*psooptim*' function within R. In conjunction with the decision to use a PSO parameter optimisation search a decision was made to write the basic parameter search code to enable partial paralllelisation of the code in Posix runtime environments in order to further speed up the model parameter search process. Schliebs wrote an initial version of the ESN parameter search code to implement both partial parallelisation and the PSO search technique. This section of work follows the recommendations in "*How to do good research in activity recognition*" (Plötz, 2010). This section of work was reported in Hunt et al. (2014).

5.4.6 *Entity description*

The entities involved in this section of work are the horse rider and the horse. In this case the horse is a built for purpose wooden model that allows multiple mounts and dismounts in a laboratory setting without having to deal with horse welfare or safety issues.

5.4.7 *Session Script*

The script asked the participant to start and finish each mount/dismount pair at the same spot in the laboratory, within three metres of the wooden horse but clear of all obstacles. Prior to each mount/dismount pair the participant clapped her hands over her head two times as a synchronisation signal (to enable the inertial data to be synchronised with the video). After each set of claps and upon mounting the

participant was asked to pause for approximately five seconds by standing or sitting still while counting 1001 ... 1005.

5.4.8 *Sensor*

The SparkFun 6DoF inertial sensor is as described in Section 5.3.4, on page 104.

5.4.9 *Data description*

The same dataset used in 5.3.6 was deliberately chosen to enable a comparison between LSM and ESN technology and consists of one of the two laboratory riding sessions, utilising a single participant mounting and dismounting a wooden *horse* 17 times with the sensor on her right wrist, while following a scripted activity sheet.

The participant, who was an experienced rider, wore the sensor on her right wrist using a simple stretchable Velcro® bandage for attachment. The participant self-described herself as right-handed. The "*horse*" used during the laboratory sessions was a built-for-purpose wooden framed horse (Diana) of approximately 16 hands in height (163cm at the "*shoulder*"), draped with a standard European riding saddle and stirrups.

During the laboratory sessions the video camera was fixed into position using clamps so that the researcher was free to move around if needed. Again, figure 5.5, on page 106 shows a sample of the data from two mounts and dismounts recorded during this session.

5.4.10 *Input Data Editing & Error Checking*

This was as described in Section 5.3.7 on page 107.

5.4.11 *Data Synchronisation & Labelling*

This was as described in Section 5.3.8 on page 107.

5.4.12 *Data Cleaning and Pre-processing*

The sensor readings are normalised to a range between -1 and 1. The ESN prefers input in the range $[-1, 1]$ and so normalisation gives an opportunity to extract maximum information from the signal. Prior to normalisation the 0.1% upper and lower quantiles were removed as suggested in *Discovering knowledge in data: an introduction to data mining*, Larose (2014, p. 35). Removing the outliers and replacing them with the signal mean value allows for the maximum signal range at the cost of a very small number of changes to the signal.

5.4.13 *Data Encoding*

One advantage of using an ESN model without LIF spiking neurons is that the input data does not need to be encoded into spike trains. The time series data obtained from the sensors are cleaned, pre-processed (as above) and presented to the reservoir in the form of an ordered sequence of real-valued data vectors.

5.4.14 *Reservoir Description*

The Reservoir is a transformation of the inputs into a high-dimensional, intermediate, feature space that is then reflected by the output neurons at each time step.

The three components of the ESN model are the Input, Reservoir and Output neurons. One input neuron is employed for each input data vector (9 in this case), one neuron for each output Class (2 in this case) and a parametrised, selectable number of Reservoir neurons (939 in this case).

Table 5.2: Set up parameters for the initial ESN model.

Fixed Options		
Item	Description	Justification
Neural model	Analogue sigmoid activation using $tanh$	Jaeger (2005)
Input connectivity	Fully connected, unidirectional, input to reservoir, random distribution of weights, approximately 50% excitatory and 50% inhibitory (random sample -1 and $+1$)	Jaeger (2005)
Reservoir connectivity	Fully connected, asymmetric topology between reservoir neurons, random, normal distribution of weight values with a mean of 0 and sd of 1	Jaeger (2005)
Variable, Run-Time Options		
Parameter	Description	Justification
Input values	9 vectors of normalized, real-valued data	Input
Number of neurons	939	Parameter search
Input scaling	0.9147	Parameter search
Neuron leak rate	0.0594	Parameter search
Linear Regression Regularisation	2.7000	Parameter search
Spectral Radius	1.0000	Parameter search

In a generalised ESN model, each input neuron is connected to each Reservoir neuron, but not directly to the Output neurons. Connections from the Input neurons directly to the Output neurons are an optional feature of ESN's. Each connection from the Input to the Reservoir has a randomly set weight of either -1 or +1. This results in approximately 50% excitatory and 50% inhibitory connections.

Each Reservoir neuron is connected with each other Reservoir neuron, including itself. Each Reservoir neuron interconnection has an initial value taken from a random, normal distribution with a mean of 0 and a Standard Deviation of 1. These initial Reservoir weights are then scaled by the selectable Spectral Radius parameter value to ensure that the Reservoir weight matrix is less than unity.

The Input and Reservoir weights are initiated prior to training and remain static throughout training and testing. A subsequent reinitialisation will result in a different, random set of weights being set.

The Output neurons, in this implementation of the ESN model, are not connected back into the Reservoir. This is an optional feature of the ESN model. The Reservoir neurons are connected to the Output neurons though, of course, and the weights for these connections are initially set to zero and are then modified during training. The Reservoir neurons use $\tanh$ to implement a sigmoid activation function. In this implementation, Regularised Regression is used to train the output weights.

5.4.15 *Results*

An evolutionary algorithm, i.e. a PSO is used to do a search of the parameter space for the ESN, to find a suitable configuration to use. The regularization parameter α, the number of reservoir neurons N_x, the scaling factor of the input weights, the leaking rate a and the spectral radius $\rho(\mathbf{W})$ are optimised. The ranges for these ESN parameters are generally accepted sensible ranges and were taken from Lukoševičius (2012) and are shown in table 5.3. The lower bound for the regularisation parameter was set at zero (minimum) and the upper bound was set at 5 as a combination of the advice from Lukoševičius and experience. Lukoševičius (2012, p. 10) cautions that setting regularisation too high may result in the ESN model being too sensitive to the input (over-trained) and/or unstable but does not suggest an upper-bound. In some early attempts at optimising this parameter it tended not to go much higher than 3 or 4 and was often much less than this and so an upper bound of 5 was set and kept throughout this work.

The PSO was run with 14 particles (Bendtsen., 2012) over 50 generations and within each generation the ESN was initialized and trained five times to take some account of the stochastic nature of the initialization process. The reservoir is initialised with randomly assigned weights and so, as pointed out by Lukoševičius (2012, p. 8), each reservoir's performance can vary slightly even when the meta-parameters are the same. Lukoševičius suggests that for "large" reservoirs that this variation in performance is minor but does not, specifically, define what is meant by "large" although Jaeger (2005) suggests that anything over 100 neurons is large for an ESN. The author's experience supports this supposition that performance variation is minor and this is supported by the overall performance of the classifier when presented with

previously unseen data. However, to counter this variance in the earlier grid based meta-parameter search process when compute processing took an inordinate amount of time each reservoir was regenerated three times for each test and the error cost that was used for tuning was averaged. In this iteration the PSO search and parallelisation of the code has reduced the compute resources and so each individual test during meta-parameter tuning is repeated 5 times and the error cost is averaged. In addition, as recommended by Lukoševičius the random seed is set at the start of each set of tests to aid with repeatability.

50 generations was chosen as the stopping point for the PSO throughout this work based on experience. Early attempts to optimise similar parameter sets showed that the search stabilised within 15-25 generations and looking at Figure 5.8 it can be seen that this run stabilised after around 20 generations. Each ESN was trained on the first 50% of the time series and the entire series was used for testing. The root mean square error between target and actual network output on the test set was used as a fitness measure for the PSO.

The results of this search process can be seen in figure 5.8. As is visible from the diagram, four of the five parameters (Number of neurons, Input Scaling, Leak Rate and Spectral Radius) had settled into a small range within 20 generations. Regularization, the fifth parameter, appears less critical for this problem and a range of configurations reported satisfying test errors.

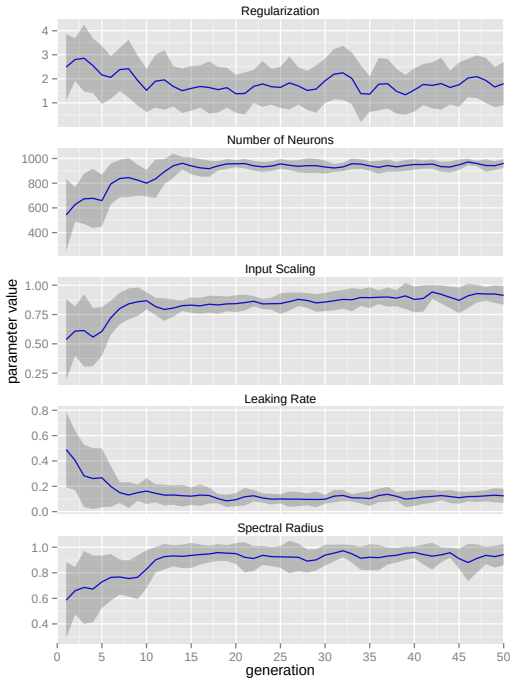


Figure 5.8: Optimising the ESN parameters

The selected parameters from the PSO that were used to run the ESN were Regularisation (2.7), Neurons (939), Input Scaling (0.9147), Leaking Rate (0.05937) and Spectral Radius (1.0). Fifty percent of the data file was used for training and the full file was used for testing. The file was divided in two parts with no account for where

ESN parameters & ranges			
Parameter	Id	Range	Justification
Regularisation	α	0 to 5	Lukoševičius (2012, p. 10)
Number of Neurons	N_x	100 to 1,000	Lukoševičius (2012)
Input Scaling		0 to 1	Full range
Leaking Rate	a	0 to 1	Full range
Spectral Radius	$\rho(\mathbf{W})$	0 to 1	Lukoševičius (2012)

Table 5.3: PSO Parameter Ranges

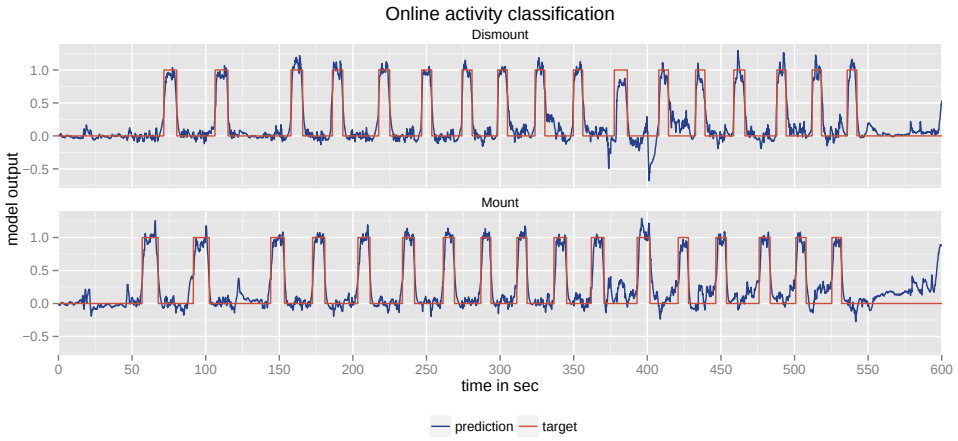


Figure 5.9: ESN Results, truncated to emphasise classes

that separation point fell in terms of the classes. In this case, as a result of where the mounts and dismounts occur, the first half of the file includes the first 8 mounts and the first 7 dismounts. The output from the test run is shown in figure 5.9.

Using a cut off point set at 0.5 of the continuous output $\mathbf{y}(t)$ has all mounts and dismounts successfully classified with one false positive mount classified during the sequence when the participant takes off the sensor (not visible in above results figure). No false positive dismounts were classified. Within each mount and dismount there is a slight lag between the class label and the classified label in most cases with an associated drop off towards the end of each class. This is an expected characteristic of the reservoir property as the reservoir needs some time to establish a recognised pattern.

5.4.16 *Discussion*

The initial goal was to develop the ESN code and re-run the LSM Realistic Data iteration so that it was possible to compare LSM technology with ESN technology. However, this was not done in a formal sense. The ESN classifier had worked well

Class	Predicted Label		
Label	0	1	2
0	3298	53	45
1	103	1281	0
2	58	0	1163

Table 5.4: Confusion Matrix of Results

in this particular situation, using data from a scripted activity in a laboratory setting collected from a single participant. The false positives towards the end and in some cases midstream, possibly indicates that the ESN classifier would benefit from additional training that provides a wider variety of signals that are outside of the desired classes.

It is worth noting that the use of overhead claps for synchronisation and the scripted pauses in activity introduce noticeable signal artefacts into the data. While it is not possible with an ESN to tell what properties of the signal are used by the classifier it is probable that the artefacts make a contribution to classification.

For example, the initial spike in the classifier response in the wider gap between the second dismount and the third mount (at around 125 seconds in Figure 5.9 on page 119) may be indicative of the ESN starting to respond to the synchronisation signal during the slightly longer pause. In addition, the clear pause in activity after mounting consistently occurs just prior to dismounting and so this artefact may well contributing to dismount classification success. It would be unwise to conclude that the current classifier is suitable for use on data that does not contain artificial artefacts.

The current pre-processing techniques that were used to normalise the signal between -1 and +1 will have created additional signal artefacts including magnifying the drift in the Magnetometer and Gyroscope data. The removal of outliers has enabled the central portion of the signal to be amplified but the outliers are real data and so by removing them the underlying activity signatures have been changed. These issues, without resolutions, will make it difficult to generalise the results across participants and across other sensors.

This data was collected from a single individual using a single (wooden) horse on the same day using the same equipment. This provides unrealistic consistency. In the real world not only do riders not follow a script when mounting and dismounting they also come in all sizes, temperaments and skill levels; their horses come in all sizes, temperaments and training levels; additional equipment may be involved such as a rider holding a crop while mounting and differing techniques may be used while mounting. All of this cautions the author against simplistically concluding that he is close to having a simple, reliable technique of classifying this (or any other punctual activity) that works across riders and situations.

This dataset represents an idealised scenario and provides an excellent chance of successful classification. The positive results are pleasing and they need further development before it can be safely concluded that RC techniques in general and ESN

techniques in particular are suited to classifying punctual human activities based on inertial data. However, despite this caution, these results are positive, seem to be as accurate as LSM techniques and are less resource intensive than the prior Python based LSM tool-set. As a result there was a willingness to follow this path further and a general confidence that ESN techniques are as effective as the earlier LSM techniques in this area.

While the same data was used, similar results were achieved and this work was reported in a paper (Hunt et al., 2014), no attempt was made to actually compare the computing resources between both RC techniques. Instead a pragmatic decision was made to continue further work using ESN technology based on:

- the author's (developing) better knowledge of R as a programming language versus Python
- The more resource efficient parallelised version of the PSO parameter search optimiser versus the grid-based Python search optimiser that enables us to experiment widely without worrying about lengthy jobs running over numerous days that may fail midstream
- The ready availability of library packages for Machine Learning and supplementary technologies within R

The earlier LSM work was done using Python based code developed by Schliebs for Knowledge Engineering and Discovery Research Institute (KEDRI) and is part of KEDRI's tool-kit for work of this nature. This system was run on a Macintosh computer. In the beginning, the Python code was run by Schliebs while this researcher contributed by running the grid search to optimise the LSM parameters, produced some of the graphics and pre-processed the inertial data on a MS Windows based PC. The plan was to progressively move all technical work from Schliebs to the author as the author learnt how to operate and modify the code base.

At the beginning of this work, the author had a passing knowledge of programming in Python but did not have any substantive development experience in Python beyond some simplistic learning exercises and had no experience in R of any sort. For this part of the work, Schliebs wrote an underlying library routine to implement ESN techniques within R, while the author developed almost all of the ancillary R code needed to run each design iteration (using the ESN library routine), pre-process the data, optimise the model parameters and visualise the results. Note that the ESN written by Schliebs includes some plotting routines that allow the ESN model to be visualised. As a result, the author learned R and learnt it to a depth beyond his knowledge of Python.

The results produced using the ESN model were comparable to those produced from the LSM model with similar performance to the Python code. In addition, it became clear that R is a very powerful language and environment for scientific computing with libraries for all of the areas needed in this investigation. Schliebs then left the research team. With substantive time invested in learning R versus a passing knowledge of Python and greater familiarity with the process of operating the ESN routines it was decided that ongoing research would be conducted using R

and that meant that a decision would need to be made about how to use LSM models in future research, the options seemed to be:

- Interface R with the Python code when needing to run the LSM code
- Re-write the LSM code in R
- Stick with using ESN models only for the remainder of this research.

With the departure of Schliebs the author no longer had ready access to a Macintosh based computer and had reduced access to the LSM code written in Python. While both of these issues could have been resolved they involved effort and additional resources that were not absolutely necessary.

Without access to both the LSM and ESN technology on the same hardware platform it would have been difficult to quantitatively assess the performance of one technology against the other. While putting them both on the same hardware platform was possible, it required time and effort that was better spent further investigating one of these two branches of RC.

A start was made along several lines before coming to this conclusion. Firstly a Macintosh computer was borrowed and an attempt was made to install and run the LSM Python code but the author's lack of experience with both Mac OS and Python meant that this process was time consuming and frustrating. The author also considered running the ESN R code under Mac OS and this was probably the best of the choices in terms of performance via a single computer but ultimately it was decided that relative performance on a single computer was not an important enough issue to invest time into resolving when the possibility existed of running compute intense parallel processing optimisation runs on clustered Linux computers instead.

5.4.17 *Results versus Goals*

1. that the newly written ESN R code worked as planned by achieving comparable results vis-a-vis the Python based LSM code.

   The core ESN model code went through two major iterations before the author was happy that it implemented a ESN model with suitable functionality. Both versions were written by Schliebs with some feedback from Hunt. The biggest difference between the versions is that the initial version was written using well structured for loops for repetitive functions while the second used R's matrix operators and thereby enhanced possible future parallelisation attempts. This use of R's matrix operators had a tendency to decrease execution times, although this was not measured formally and nor was any performance increase particularly notable.

   In addition, the initial implementation allowed for both real and discrete outputs as a selectable run time parameter. Between the versions it was decided that this was outside of the core functionality required from this code and so the code was altered to allow for real outputs only, with the possibility of discretising those outputs, when required, using specific functions for that as post-model processing activities.

The third major change was that model plotting routines that were initially included within the core code were extracted out and provided as a separate function to enable changes to the plotting code to be kept separate from the ESN model creation and running routines.

Lastly, the initial code included techniques for undertaking parameter searches using both a grid search and PSO search. This code was also taken out and placed into a separate ”tools” routine in the second version of the code.

Schliebs undertook additional tests and declared the code ”complete” and ”functioning correctly” as per Lukoševičius and Jaeger (2009), then handed the code over to Hunt. Hunt accepted the code once it became apparent that it was achieving comparable results to the LSM iteration.

The 500 neuron LSM model using parameters optimised via a grid search processed the LALab file and correctly classified all mounts and dismounts with false-positive mounts and dismounts at the end of the input, using a 0.6 cut-off. Sample by sample testing classification using the LSM was 85.1% correct in this instance.

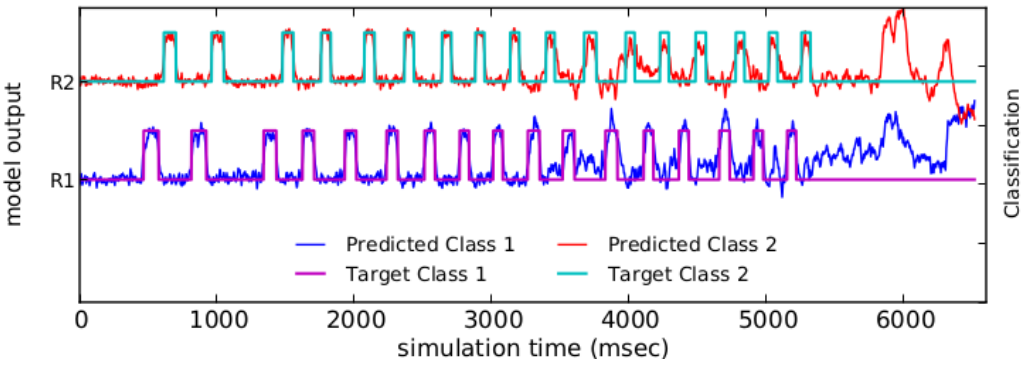


Figure 5.10: Non-truncated results of LALab classification using LSM.

The 939 neuron ESN model using parameters optimised via a PSO search processed the same LALab file and also correctly classified all mounts and dismounts and also produced false positive mounts and dismounts at the end of the input using a 0.5 cut-off. A 0.5 cut-off being more sensitive to false-positives. Sample by sample testing classification using the ESN was 95.6% correct in this instance. It was interesting to note that the output signal from two different RC techniques appear to be very similar, including the signal shape at the end where the rider is taking off the sensor.

The stochastic nature of ESN models is demonstrated in Figure 5.11. Producing this figure meant re-initialising the model as the initial run had only produced and retained the truncated figure and when the model was re-initialised the input and reservoir neuronal weights were reset. Even though the same parameter values were used for this model it has an additional false-positive mount just after the eleventh real mount, undoubtedly as a result of differing internal neuronal weights. A different view of this output data showing just the classes at the 0.5 cut-off level is shown in Figure 5.12. This view makes it easier to see where the false-positives occur.

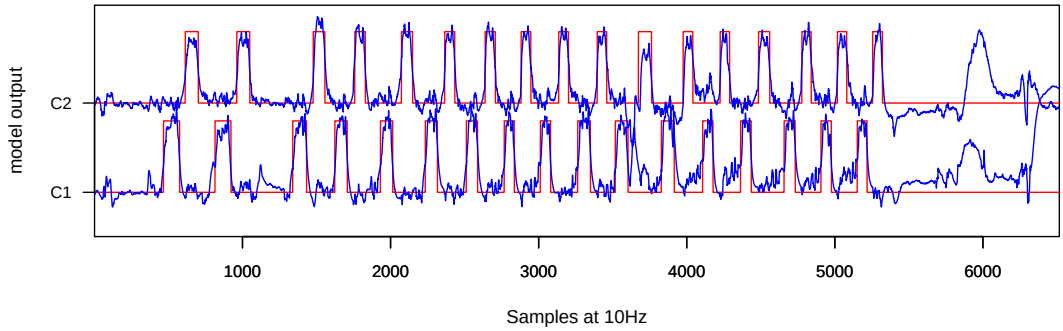


Figure 5.11: Non-truncated results of LALab classification using ESN.

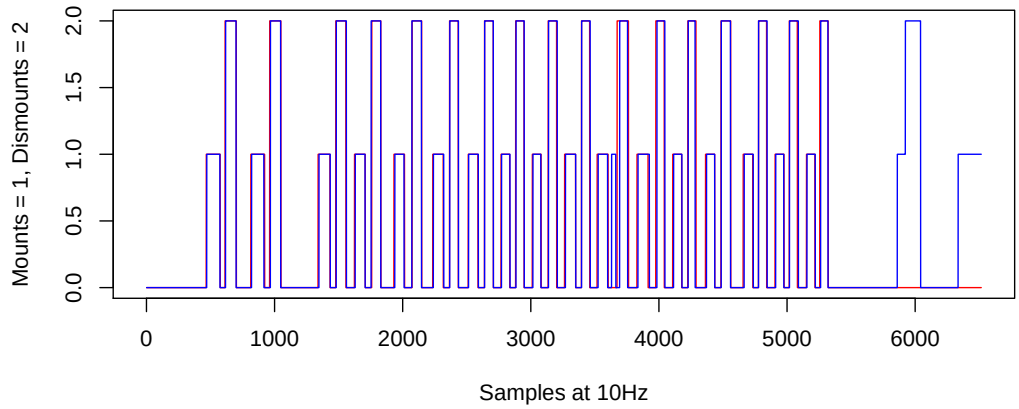


Figure 5.12: Classes from LALab classification using ESN, at 0.5 cut-off.

It is suggested, based on these reported outputs, that the ESN R code has worked as designed and has achieved comparable results to the Python based, LSM code and as a result this design goal has been achieved.

2. that the partially parallelised version of the PSO search ran with considerably less elapsed time than the grid-based search, in an appropriate environment.

   A single grid based search for the Python based LSM code typically took five to seven days (120–168 hours) of elapsed processing time to complete on a single MS Windows based i7 PC. To take account of the stochastic nature of this LSM code each parameter set was run three times and the error rate was averaged across the three runs. A similar grid based search was initially written in R for the R based ESN code. These R encoded grid based searches took a similar amount of time to run, typically around seven days, on a MS Windows based i7 PC.

   After writing the PSO R based code a search across a similar parameter set on the same MS Windows based i7 PC completed in hours rather than days. On the same MS Windows i7 PC a non-parallelised PSO parameter search

took 20 hours, 38 minutes and 26 seconds elapsed time to run with five repeats of each ESN being run and averaged, for 50 generations versus only three repeats when using the Grid search. On a VMWare virtual eight core machine running Debian the same search, across the same data, again with five repeats and 50 generations, took 5 hours, 55 minutes and 6 seconds elapsed time to run. Table 5.5 summarises these results.

The reduction in elapsed time with the parallel PSO search on the Debian machine was noticeable and was a major factor in deciding to stick with R, ESN models and PSO parameter searches. With this reduction in elapsed time it became feasible to test multiple scenarios without losing significant amounts of time for each test, especially if a test failed for some reason and needed to be repeated.

Search	Machine	Op Sys	Parallel	Repeats	Elapsed Hr:Mn
Grid	i7	MS Windows 7	No	3	144:00
PSO	i7	MS Windows 7	No	5	20:38
PSO	Virtual 8 core	Debian	Yes	5	5:55

Table 5.5: Comparing each parameter search techniques elapsed time performance

R and RStudio have shown their versatility and usefulness in many situations. For example, when more sophisticated graphing was needed, the GGPlot2 library that is powerful and (relatively) simple to use was found. When a SVM or Random Forest model was needed as a classifier for the same input data, a R library to do this was readily available and relatively simple to implement. R code can be interactively developed and run within RStudio on either a local PC running MS Windows, on a Macintosh computer or via Amazon's Elastic Cloud Compute facility, utilising Debian multi-processor virtual machines over the Internet when required. R's flexibility and usability both within a MS Windows environment and across other operating systems environments has been exemplary. With the addition of Latex and GIT, R and RStudio provide a powerful researcher workbench for research of this nature.

R code development has continued and Hunt is now the only researcher working on code development for this research and so Ipso Facto, Hunt has learned enough R to take over R development for this research.

In addition, Hunt et al. (2014) and Hunt and Parry (2015) demonstrates that Hunt has been able to take on the role of lead author for papers and to lead the research.

The parallelised version of the PSO code uses the R package '*parallel*' (R Core Team, 2012) to achieve parallel processing. This library function utilises '*fork*' to run multiple threads on multi-core processors under POSIX environments. Under MS Windows environments there is no equivalent to 'fork' and so the same code runs as a single thread. Running the same parallelised code

in both Linux and MS Windows environments results in the same outputs but with shorter elapsed run times under multi-core Linux environments.

5.5 AMENDING THE ACTIVITY DEFINITION - MOUNTING V MOUNTED

After running the realistic data iteration against the LSM model, as described in section 5.3 it was realised that while the LSM (and later the ESN) did indeed normally take some time to ramp up to a correct activity classification at the start of the activity, that this ramp up period was usually short and in the order of one or two samples. As a result the decision to add a 10 sample start buffer to the activity class was questioned. At the same time, the decision to add an end buffer to encapsulate some small part of the *mounted* activity that follows *mounting* was also questioned.

Lastly, while looking at the activity definitions the wisdom of starting the stirrup mount when the rider has both feet on the ground, versus at a later point was questioned. With the laboratory activities that used a 'wooden' horse questioning this start point versus a later start point was mute as the 'wooden' horse did not move during mounting. In the real world, however, riders mount real horses rather than 'wooden' horses and real horses sometimes move away from a rider as they try to mount. Without going into a longish explanation about why this might be, let it suffice that this moving away from the rider is encountered often enough to be worth considering. Within the real-world data, there are real examples of this behaviour.

When a horse moves away from the rider as they prepare to mount, the rider must follow the horse, let the horse go or continue holding on while the horse moves away and risk overbalancing and falling on the ground. Typically, riders encounter this behaviour after they have put their left foot into the stirrup and so if they hold onto the horse as it moves they end up hopping along beside the horse as it moves until they are stable enough to start mounting.

In one session, the rider hopped for several minutes before being able to successful mount. As a result, sticking with the initial definition of mounting both makes mounting different between situations when the horse moves (and it does not move) and increases the time variance between different situations. In addition, using the last time the rider has two feet on the ground results in the inclusion of false start mount data when the horse moves away enough to have the rider take their foot out of the stirrup and start again. It is suggested that this false start mount data may prejudice the ability to correctly classify mounts in real world situations.

It is also worth noting that it can take considerable effort for a rider to raise their body when doing a stirrup mount especially when a rider who is somewhat shorter than a particular size for the size of the horse attempts a stirrup mount. In this case (short rider and/or tall horse) then the rider often tends to "bounce" once or more before doing the actual mount in order to gain enough momentum to get their body up high enough to get their leg over the horse's back.

These bounces could either be viewed as a form of preparation for mounting and included or they could be viewed as a sub-activity separate from mounting and only encountered in some limited cases and so not part of the mount. This later stance was adopted and any bounces were not included in the new re-definition of *mounting*.

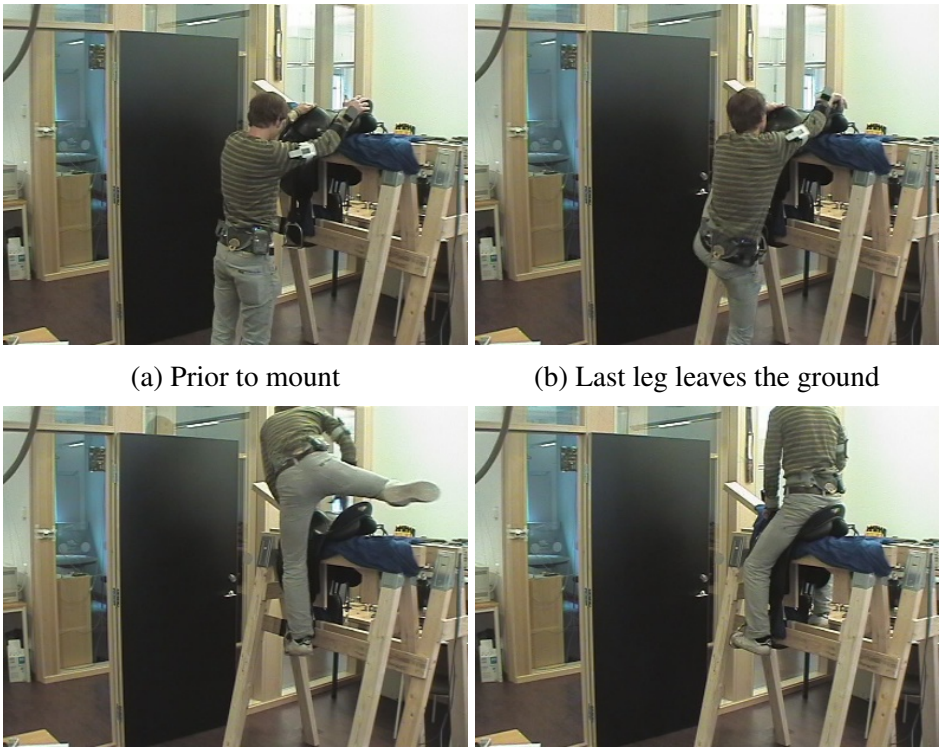
In terms of horse welfare, bouncing prior to mounting while having one foot in the stirrup puts unnecessary pressure on the horse's back, especially if the saddle is ill-fitting or badly adjusted and so can injure or hurt the horse. In general, knowledgeable riding coaches strongly discourage riders from bouncing prior to mounting and usually encourage the rider to instead use a mounting block or other stationary thing to assist in raising their body in relation to the horse in order to prevent injury to the horse while mounting.

A similar definition is proposed that starts just a little bit later (for non-moving mounts) and is more general, results in less time variation and excludes possible bounces pre-mount. It is more succinct and more general. With dismounts, particularly drop-style dismounts, the rider will often break the impact of landing back on the ground by bending their knees and then bouncing back upright. On reconsideration, it was decided to have a similar philosophy to mounts and only include the minimum aspects of dismounting and so a decision was made to end the dismount when both of the rider's legs first touch the ground. Thereby excluding any knee-bend or hop on dismount. For both activities it was also decided to remove any buffer from both ends and rely on the classifier responding quickly enough to capture the event.

As with all activities, identifying the precise beginning and ending of a mount/dismount is not trivial, this is supported by other researchers such as Plötz (2010). Both the prior definition and the new definition are difficult to identify exactly within this dataset as the author had not defined the mount (or dismount) activities formally at the time the data was collected. Without a pre-defined definition for the activities of interest the author did not know exactly where to point the camera during data capture and so, in general the camera shot records the rider's upper body but does not show their feet and so in pinpointing the start of the mount an observer needs to look on the video for an upwards body movement after the rider has put their foot into the stirrup and use that as the start of the mount. For the dismount, an observer needs to look for the frame where the rider's body first stops moving after dismounting and use as the end point. This results in a slightly arbitrary start and end points for videos that do not show the rider's feet. These start and end points are open to interpretation and can move by a couple of samples depending on each viewing of the video. At this stage it is not clear if these start and end points for the activities are useful or not useful and this needs further investigation and thought. In the meantime, they are what was used in the following work.

5.6 FORMAL RE-DEFINITION OF A STIRRUP MOUNT

A stirrup mount is defined as the period from when *a rider with a foot in a stirrup, lifts their last foot off the ground and starts mounting until the point where they have their weight in the saddle*.



(a) Prior to mount (b) Last leg leaves the ground

(c) Mounting (d) Mounted

Figure 5.13: Mounting the Lab horse using a stirrup mount technique

5.7 FORMAL RE-DEFINITION OF A DISMOUNT

A dismount is defined as the period from when *a rider, seated in the saddle, leans forward to start dismounting until the point where both feet first touch the ground.*

Having redefined the two activities the author then went back through the videos and placed new class labels on the samples that were included within the redefinitions. A summary of a sample of the changes is shown in Figure 5.6 to give a feel for how the classes actually changed after redefinition.

While the classes were redefined after running the LSM iteration and prior to the first ESN design iteration, the author did not want to re-run the LSM iteration against the new classes because of the very long compute elapsed time needed to run the grid search for the LSM iteration and so the author ran the initial ESN iteration against the same data as the LSM iteration (Section 5.4) to enable a comparison between the two techniques and then ran the following design iteration (Section 5.8) to compare the changes to the class labels using the new ESN tools.

(a) Prior to dismount (b) Leaning forward

(c) Dismounting (d) Dismounted

Figure 5.14: Dismounting from the Lab horse using a drop technique

	Original Class			New Class		
Activity	Start	End	Length	Start	End	Length
Mount 1	5067	5175	109	5138	5157	20
Mount 2	5415	5522	108	5484	5507	24
Mount 3	5940	6032	93	6004	6023	20
Mount 4	6224	6307	84	6278	6298	21
Mount 5	6532	6614	83	6585	6604	20
Dismount 1	5214	5301	88	5235	5256	22
Dismount 2	5562	5649	88	5591	5609	19
Dismount 3	6078	6156	79	6104	6118	15
Dismount 4	6359	6429	71	6383	6402	20
Dismount 5	6674	6748	75	6693	6712	20

Table 5.6: Comparison of a sample of original and redefined classes from LALab

5.8 REPEAT ESN REALISTIC DATA ITERATION WITH MODIFIED CLASSES

The second step in this exploratory phase is to demonstrate the ability of the ESN model to classify the new, shorter classes by re-running the prior iteration with the same input data but with the new class definitions.

5.8.1 *New Classes Problem Identification*

In this case, the problem to be resolved is very clear, the definition of the two classes has changed and so it became necessary to demonstrate that the new classes could be classified at a similar level of reliability as the prior classes.

5.8.2 *New Classes Motivation*

The motivation for this design iteration was also clear, the newly defined classes could not be used without a clear demonstration that they would be classified reliably using the same data and techniques as the prior classes.

5.8.3 *New Classes Design Goals*

The goals for this design iteration are: That the activities within the LALab data can be classified at a similar level of accuracy using the new classes as that achieved using the original classes. While a similar or better level of correct classification would be desirable, a less accurate classification on this dataset will still be acceptable because the original class definitions could more easily encompass extraneous signal artefacts caused by the regularity of the laboratory script process. So, even if less accurate classification results, the new class definitions will be retained because they are more accurate definitions that will hopefully translate into a more general definition especially within the real data. A large drop in accuracy on this dataset though will be some concern.

5.8.4 *New Classes Measures of Success*

A classification rate similar to that achieved by the classifier from Section 5.4.

5.8.5 *New Classes Demonstration Description*

The Entity descriptions, Session script, Sensor description, Input data editing & error checking, Data synchronisation & labelling, Data cleaning and pre-processing and Data encoding will be as described within Sections 5.4.6, 5.4.7, 5.4.8, 5.4.10, 5.4.11, 5.4.12 and 5.4.13 on pages 114 to 116.

5.8.6 *Data description*

The same dataset as used in 5.3.6, namely LAlab, was again used. Of course the class labels have been redefined but otherwise the input data is as previously described.

Figures 5.15 and 5.16 show the input data with the original long classes and the same data with the shorter classes. The long classes are easily discernible by the wider alternating red and blue 'stripes', while the shorter classes in Figure 5.16 are harder to discern at this level of zoom.

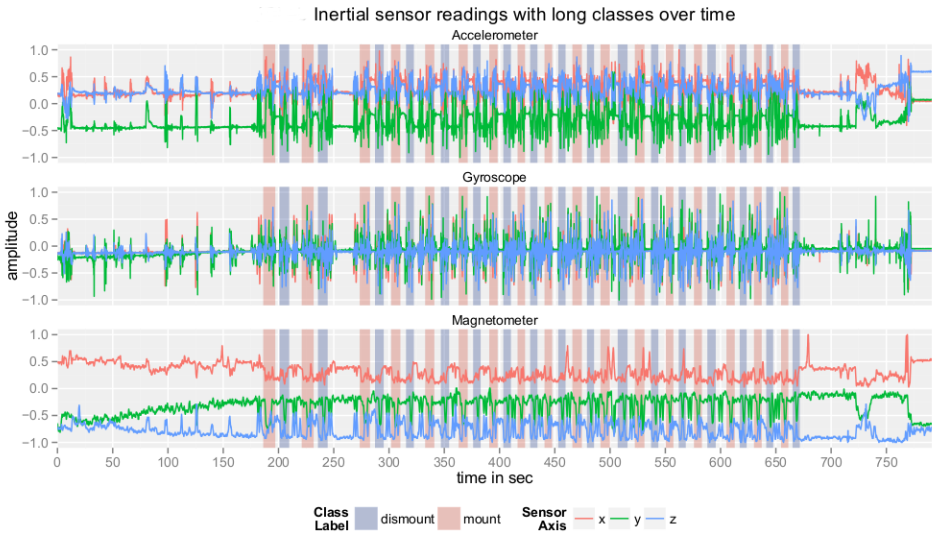


Figure 5.15: LALab input data with longer classes

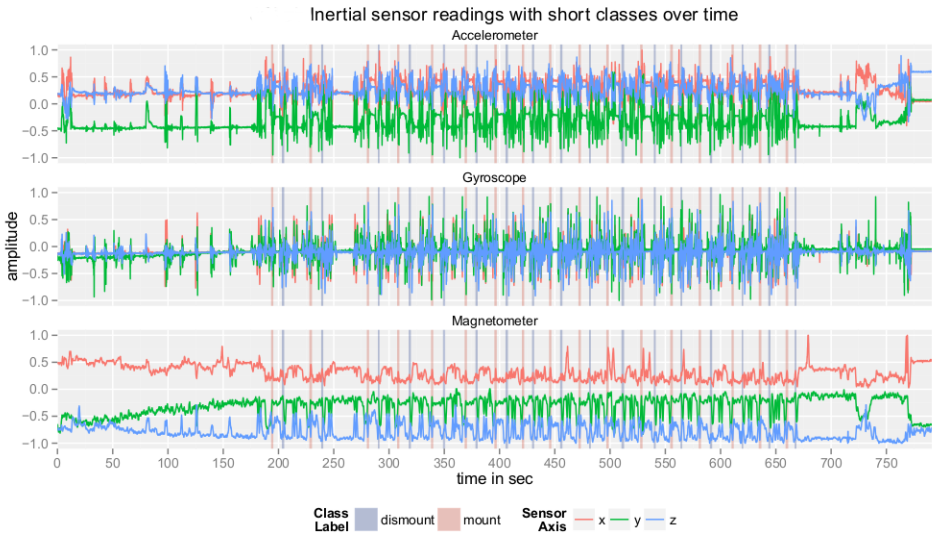


Figure 5.16: LALab input data with shorter classes

5.8.7 *PSO parameter description*

PSO parameters range for search			
Parameter	Id	Range	Justification
Regularisation	α	0.0001 to 5	S. 5.4.15
Number of Neurons	N_x	200 to 1,000	Lukoševičius (2012)
Input Scaling		0 to 1	Full range
Leaking Rate	a	0 to 1	Full range
Spectral Radius	$\rho(\mathbf{W})$	0 to 1	Lukoševičius (2012)
ESN Repeats		7	S. 5.4.15
Particles	S	14	Bendtsen. (2012)
Generations		50	S. 5.4.15

Table 5.7: PSO Parameter Ranges for short classes

A PSO search was run using the parameter ranges listed in table 5.7. This search took 6hrs 31mins 23.5 secs elapsed time to complete on an Amazon ECC virtual 8 processor machine running Debian. The ESN parameters with the lowest fitness score for the shorter classes was found in the 50^{th} generation and they are listed in Table 5.8, column (b), alongside the prior parameters used for the longer classes, for ease of comparison. A second set of short class parameters that came out of a test PSO search are also shown, in Table 5.8, column (c). With this iteration, the opportunity was taken to vary the PSO search ranges with a non-zero lower bound was added for the regularisation parameter, the lower bound for reservoir size increased to 200 and the number of repeats of each set of tests was increased from 5 to 7.

With regularisation it was noted that prior searches had spent time trying to drive the value for regularisation down to values close to zero that were almost indistinguishable from each other because of their small size and so a decision was made to limit how small this value could become so that more of the search time could be spent in more productive areas. The lower limit on the reservoir size was increased to reflect that most good solutions were found well above 500 neurons and so this also enables the PSO search to concentrate its efforts in more productive areas. Lastly, with the increase in search efficiency there was extra compute resources available and so these were used to help increase confidence in repeatability by increasing the number of times the reservoir was regenerated.

5.8.8 *ESN parameter description*

Table 5.8 (b) shows the ESN configuration that provided the minimum score from the PSO run. The PSO code progressively reports its progress and in this case the researcher tried some of the interim sets of parameters and noted one set [Table 5.8 (c)]

that seemed to provide a useful outcome despite not having the minimum found score and this alternate parameter set and its results are also reported here.

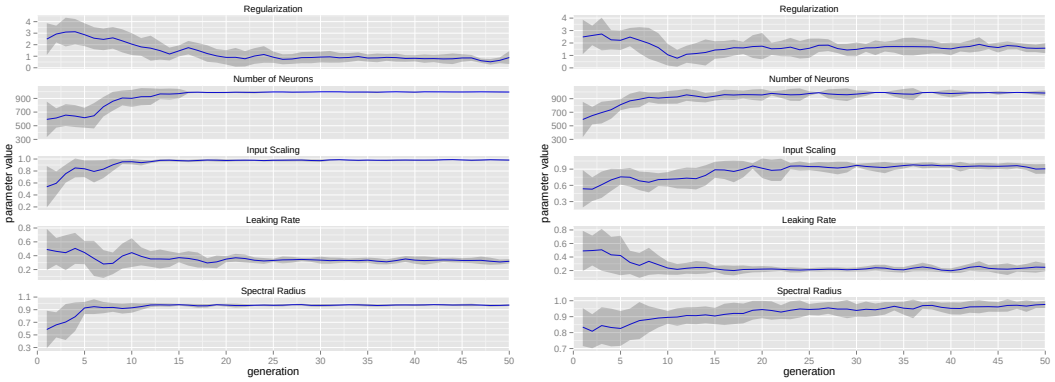
Table 5.8: Set-up parameters for the second ESN model.

Fixed Options				
Neural model	Analogue sigmoid activation using $tanh$			Jaeger (2005)
Input connectivity	Fully connected, uni-directional, input to reservoir, random distribution of weights, approximately 50% excitatory and 50% inhibitory (random sample -1 and $+1$)			Jaeger (2005)
Reservoir connectivity	Fully connected, asymmetric topology between reservoir neurons, random, normal distribution of weight values with a mean of 0 and sd of 1			Jaeger (2005)
Variable, Run-Time Options				
Input values	9 vectors of normalized, real-valued data			
	Long class (a)	Short class (b)	Alt Short (c)	Justification
Number of neurons	939	994	779	Search
Input scaling	0.9147	0.9704	0.7835	Search
Neuron leak rate	0.0594	0.2716	0.2252	Search
Regularisation	2.7000	1.1632	1.0293	Search
Spectral Radius	1.0000	0.9595	0.9343	Search

5.8.9 *Optimisation Results*

A pictorial view of the PSO search process progress is shown in Figure 5.17, (b). It is juxtaposed to the same view of the prior search progress for the longer classes [Figure 5.17, (a)] to aid comparisons.

While most ESN parameters settled after 20 generations for the longer classes, with the shorter classes the search space was still yielding parameter sets that result in better fitness scores right through until the PSO search was terminated at 50 generations. In particular, Spectral Radius (a parameter with a logarithmic effect) continues to show variation throughout the search. The number of neurons settled into a (high) narrow range after 40 generations. Input scaling and Leaking rate show some bursts of variation that may be associated with Spectral radius.



(a) Long classes (b) Short classes

Figure 5.17: Comparing the PSO outputs - LALab Long and Short classes

5.8.10 *Demonstration Results*

ESN models were run using both the parameter set from the PSO that produced the lower score and the alternate parameter set. The real outputs from these runs are shown in Figure 5.19 and Figure 5.20. in addition, the output from the ESN model that was run using the original, longer parameters is shown in Figure 5.18 for comparative purposes. The two false positives that are marked within Figure 5.18 occur when the output signal exceeds 0.6 and consist of a Mount false Positive associated with the participant removing the sensor (at around 7900) and another one at around 5000, immediately after a successfully classified Mount. In the latter case this may be some sort of ripple effect from the prior Mount as the signal amplitude drops quickly after the Mount then immediately rises quickly for the False Positive and then drops again equally quickly.

The shorter classes can be seen in Figures 5.19 and 5.20, compared with Figure 5.18. The figures have been marked with arrows to make it easier to see the raw false positive ESN outputs. At this level of zoom it is harder to pick up the false negatives in a pictorial view, but these are shown in the Confusion matrices. Again, the false positives occur when the output signal exceeds 0.6 and in this case there is only a single Mount false positive that occurs towards the end of the plot and is associated with the participant removing the sensor. There are also five Dismount false positives, two at around 5800 that may be associated with a slightly longer pause between mounting and dismounting, another at around 6400 that seems to align with a lower amplitude output at the same spot in Figure 5.18 and the last two towards the end of the plot that are also probably associated with the participant taking off the sensor. Figure 5.19 shows six segments of raw false positive output at an 0.6 score cut off level. Two of these segments, towards the end of the dismount signal, are very close together and are indicated with a single arrow.

Figure 5.20 was produced from the alternate set of short class parameters and it can be seen that there are only three segments of false positive output scores at the 0.6 cut-off level. Interestingly, these align with similar false positives towards the end of the plot in figure 5.19, perhaps indicating consistent signal artefacts in these areas.

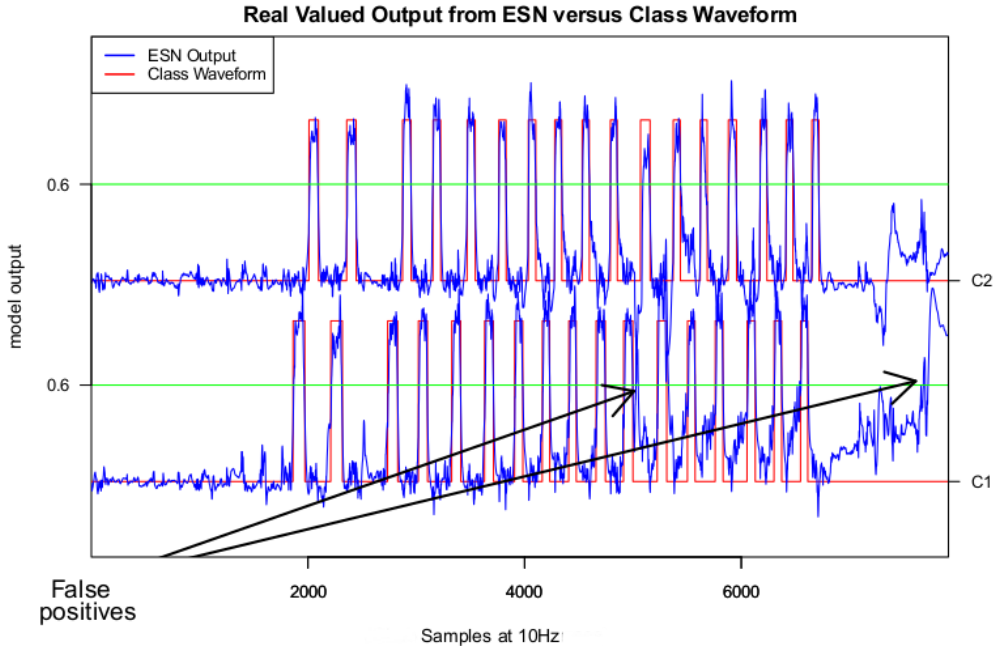


Figure 5.18: LALab ESN output for long classes

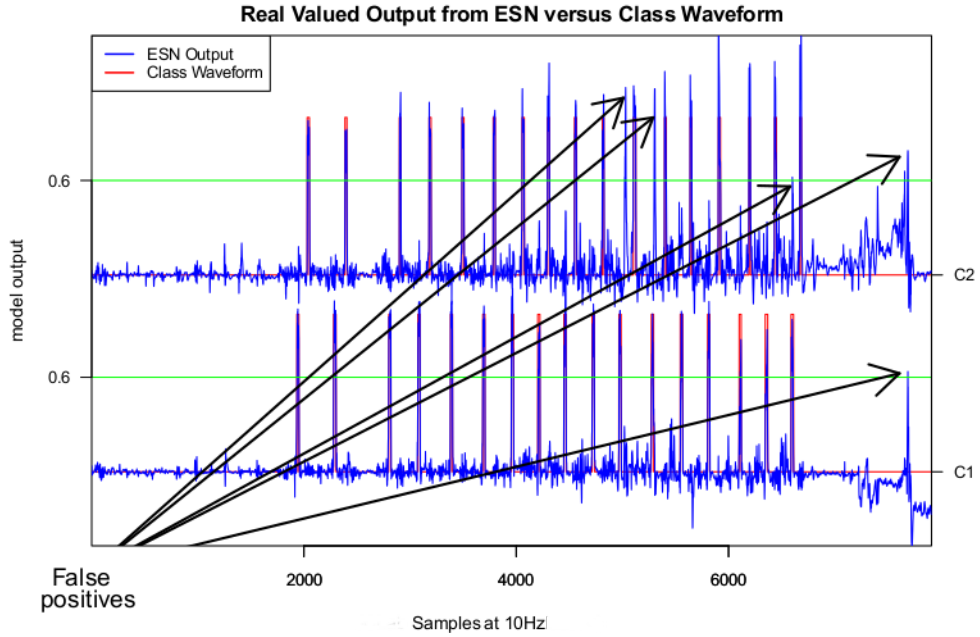


Figure 5.19: LALab ESN output for short classes

This alternate classifier seems to have a more useful outcome and this alternate parameter set and its outputs, measures and implications are discussed in the following discussion section. Both ESN models utilising the shorter classes have produce more raw false positive score segments than the ESN model optimised for the original, longer classes.

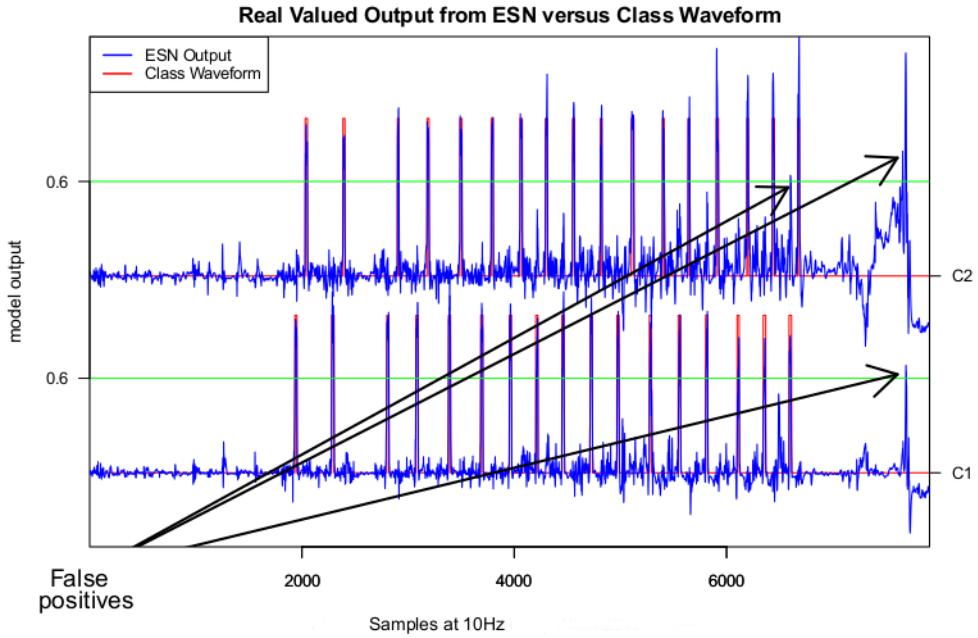


Figure 5.20: LALab ESN output for short classes, alternative parameters

The real-valued output from the ESN models is subsequently passed through a post-processing function that turns the real-values into class labels. Confusion matrices base on these class labels are shown in tables 5.9 (a) and (b) and table 5.10.

Class	Predicted Label		
Label	0	1	2
0	5036	252	19
1	148	1236	0
2	129	0	1092

(a) Long classes, minimum score

Class	Predicted Label		
Label	0	1	2
0	7189	0	33
1	80	289	0
2	25	0	296

(b) Short classes, minimum score

Table 5.9: ESN Confusion Matrices for LALab Long and Short classes at 0.6 cut-off

The change in the class width is (obviously) reflected in the increased number of samples that are in the base class versus the mount and dismount classes and this makes comparing the raw confusion matrices difficult. The difference is not great, however, and the two confusion matrices from the two runs have similar numbers. To aid comparison between the longer and shorter classes the classifier precision and recall based on multi-class precision and recall (Forman, 2003) have been calculated. These results are shown in Table 5.11.

These results demonstrate that all three ESN models did very well at classifying this data. Use of shorter classes seems to have improved both precision and recall

Class	Predicted Label		
Label	0	1	2
0	7196	0	26
1	84	285	0
2	29	0	292

Table 5.10: ESN Confusion Matrix for LALab Short classes using alternative parameters

Iteration	Statistic	Base	Mount	Dismount
Long classes, minimum score	Precision	0.9479	0.8306	0.9829
	Recall	0.9489	0.8931	0.8943
Short classes, minimum score	Precision	0.9856	1.0000	0.8997
	Recall	0.9954	0.7832	0.9221
Short classes, alternate params	Precision	0.9845	1.0000	0.9182
	Recall	0.9964	0.7724	0.9097

Table 5.11: Comparing Precision/Recall for LALab Long versus Short classes at 0.6 cut-off

for the base class by a small amount. Mounts seem to have slightly better precision but somewhat worse recall using shorter classes while dismounts are the opposite with slightly worse precision and marginally better recall using shorter classes. The author suspects that these seeming improvements are not statistically significant and so calculated the unweighed and weighed Cohen Kappa scores to test for significance.

Using the 'cohen.kappa' function from Revelle (2014) in 'R', at the default 0.05 probability level for the confidence intervals, the unweighed and weighted kappa levels of both the (a) and (b) sub-tables of table 5.9 and of table 5.10 were calculated to allow for further comparison of the classification of longer and shorter classes. This shows more clearly that the classifiers have very similar Cohen Kappa scores, implying that the classifiers are equivalent and the confidence levels overlap in all cases, giving some confidence that the classifier accuracies are similar at a statistically significant level.

	lower	est.	upper
unweighed	0.85	0.86	0.87
weighted	0.87	0.88	0.90
Number of subjects		7912	

(a) Long classes, minimum fitness

	lower	est.	upper
unweighed	0.87	0.89	0.91
weighted	0.88	0.90	0.92
Number of subjects		7912	

(b) Short classes, minimum fitness

Table 5.12: Cohen Kappa and Weighed Kappa correlation coefficients and confidence bounds that compare the classification against the known classes

	lower	est.	upper
unweighed	0.87	0.89	0.90
weighted	0.88	0.90	0.92
Number of subjects		7912	

Table 5.13: Cohen Kappa and Weighed Kappa comparisons - LALab Short Classes, Alternate Parameters

Plots of the output classes for all three cases in Figures 5.21, 5.22 and 5.23 are shown so that it is easier to visualise the final classification results. Again the false positives are highlighted with arrows on the figures.

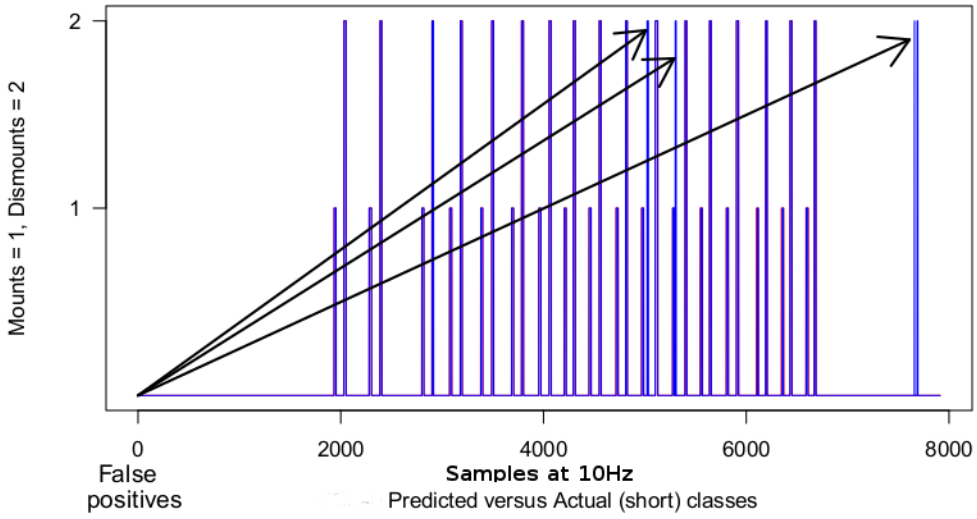


Figure 5.21: LALab Predicted versus Actual Short classes, parameters from minimised score

Here it can be seen that the two closely adjacent false positives segments highlighted in the ESN real output have been merged by the post-processing function in

Figure 5.21 and that two false positive segments that were too short for the classification criteria were filtered out.

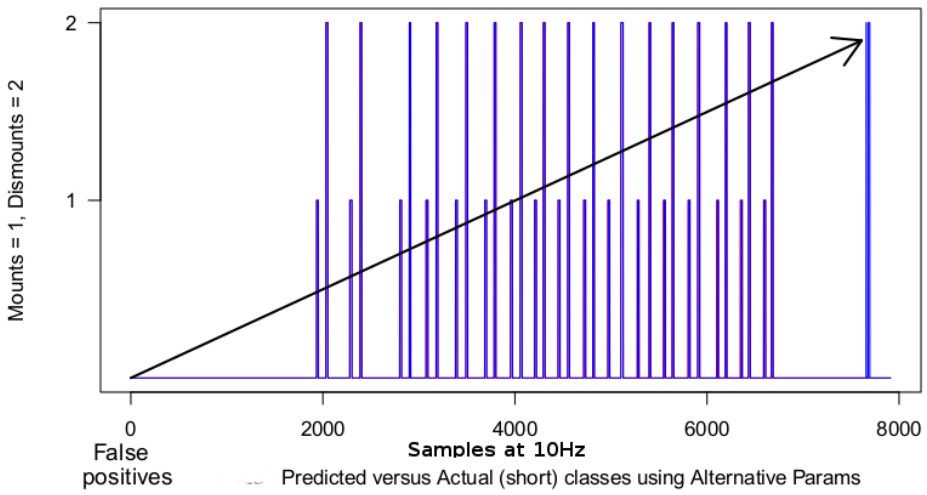


Figure 5.22: LALab Predicted versus Actual Short classes, alternative parameters

Figure 5.22 shows that two segments of real valued data above the 0.6 cut-off have again been filtered out by the post-processing, leaving only a single false positive classification segment. Figure 5.23 shows that neither set of real valued output from the longer class ESN model above the 0.6 cut-off has been filtered out by the post-processing function and so both false positive segments remain.

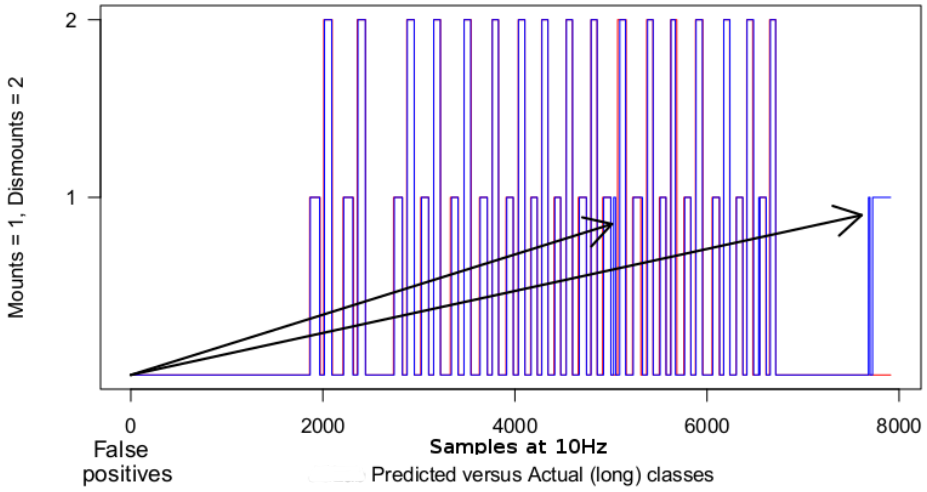


Figure 5.23: LALab Predicted versus Actual Long classes

5.8.11 *Discussion*

The raw, real-valued outputs from the ESN models seemed to be worse for both sets of short label model parameters compared with the model utilizing the longer classes and optimised parameters at an 0.6 cut-off level. However when the raw values were passed through the function that converts raw outputs into classes the precision and recall levels show that all three classifiers had similar levels of accuracy. The Cohen Kappa scores (weighed and unweighed) for all three models all fell within a 90% confidence level range for each other and conclude from this that the three models were all similarly effective at classifying the activities of interest.

It is noteworthy that the alternate set of parameters for the shorter classes produced results that were at least equally effective at classifying the activities of interest, even though those parameters were excluded from the preferred parameters during the PSO search process. This leads to questioning the effectiveness of the fitness score that is currently used within the PSO search process.

To date a single dataset, captured from a single participant under laboratory conditions that utilised scripted activities has been used. The same dataset has been maintained throughout to better enable the ability to compare model changes (from LSM to ESN) and class definition changes (from longer to shorter classes). There is now a need to test models based on more than one dataset and more than one participant to see if the levels of classification reliability remains.

The work so far has split the dataset into two parts, one for training and another for testing and the testing dataset includes all of the training data. These same two datasets have been used both while searching the parameter space for optimised parameters and also while testing models based on those parameters. Again, this practise was maintained to aid the ability to compare between the initial work and the subsequent work but the author's knowledge of classification best practise has improved over this time and it is now realised that using the same files during both model tuning (selecting an optimised parameter set) and model testing biases and probably over-fits the model to the datasets used (Flexer, 1996; Henery, 1994). The final tests need to be done using data that has not been seen before. In addition, the final test should not include data used during training as this tends to suggest that the model's classification accuracy is higher than it really is.

As noted in the Data chapter, the Magnetometer sensor readings are both very noisy and are dependent on geographical position in their raw form. In the work to date the issue of noise has been ignored and as a result of using a single participant in a fixed geographic position following a provided activity script the issue of geographic dependence has been irrelevant. The future work will be with files that are not subject to these constraints and so these issues need to be dealt with. Following the principles of Occam's razor, the simplest solution within the Black-Box approach is to exclude the Magnetometer sensor data from the studies from now on.

5.8.12 *Results versus Goals*

It is concluded that the design goals for this stage of the research have been met as it has been demonstrated that two ESN models were able to classify Mounts and Dismounts at equivalent levels of accuracy to the original ESN model that was based on the longer classes.

5.8.13 *How the design iterated*

The next step in the design cycle is to create an ESN model using similar techniques but utilising the unscripted, real-world data, captured from a number of different participants to demonstrate that the results with the first participant were not a fluke. The following work will not be directly comparable as the items noted as better practice in subsection 5.8.11 will be implemented in order to improve the reliability of the results and conclusions.

5.9 OVERALL SUMMARY OF LEARNINGS AT THIS POINT

The author developed a set of tools that formed a workbench for future work and learned a number of things during the work described within this chapter that are relevant to the ongoing, iterative development of the classifier artefact that is the goal of this work and these are listed here because they help direct the following work. These learnings are:

- Often ESN models built using just Gyroscope data have similar accuracy to a model trained on both Gyroscope and Accelerometer data and the model utilising less input data is less resource intensive.
- Generally larger models, particularly those larger than 1,000 neurons, seemed to improve classification but this is usually as a result of over-training and when such large models are tested on new data they resulted in worse results.
- Design iterations with less sensor inputs (e.g. only using Gyroscope data) encounter over-training with less neurons and useful results can be achieved with models of around 300–400 neurons. This is consistent with the recommendations of Lukoševičius (2012) that reservoirs only be as large as needed.
- Adding small numbers of neurons to a model can improved classification but often only marginally and on a linear scale whereas the subsequent increase in resource usage is logarithmic.
- Some quite small models with 50–60 neurons can achieve quite useful classification in some circumstances and appear to be more sensitive to other ESN model parameters. This is consistent with both the advice of Lukoševičius (2012) and the suggestion by Jaeger (2005) that smaller models, below 100 neurons can be used as some form of ensemble.
- For data that is relatively symmetrically distributed around the zero mean (as with Gyroscope data) improvements in classification can be achieved in some

cases by adding a linear offset to the data. See also Jaeger (2005). In these cases adding a constant input that is offset from zero sometimes has a similar effect.

- Using a simplistic error cost function that calculates the normalised mean square error across the entire model output during parameter tuning can result in parameters that are optimised to drive the model output towards zero when used with data where the class values versus non-class values are very rare. Within this work, class values typically number around 15–22 across a typical real world data file containing 36,000–50,000 rows and so the class values are rare compared with the non-class values.
- Generating an ESN model for a single class (e.g. Mounts only) substantially reduces the computing resources needed to compute the machine learning model.
- Working with a two class classifier (base class and one activity class) means that it is easier to compare between ESN models as most existing performance measures such as area under the ROC curve are designed for two class problems.

Chapter 6

REAL WORLD DATA DESIGN CYCLE

This chapter is the third phase of classifier development and involves using the Researcher Workbench and RC classifier tool-set that was developed during the prior chapter to iteratively create a RC based classifier that is capable of classifying the real-life activity of interest. This process results in the development of the artefact that meets the overall research goal.

6.1 INTRODUCTION

Chapter 5 describes the overall development goals and the first, proof-of-concept, phase of this work. This is followed by a summary of the work done exploring different facets of the data, the ESN techniques for classifying it and the development of the tool-set into a researcher's workbench. The earlier work gives confidence that ESN techniques are applicable for classifying spatio-temporal inertial data and the exploration work provides experience with differing ways of applying ESN techniques to this data.

This chapter brings that earlier work and the learnings from that work (as summarised in Section 5.9 on page 141) together in a focused attempt to classify the real world data. In the earlier work it wasn't clear if classification was succeeding because of inherent properties of the activities of interest or instead because of the properties imparted to the data as a result of the participants following a regular, artificial script. In working with real-world data the author expects that the data will be much less regular and so there is more likelihood that successful classification will be as a result of the inherent effects of the activity of interest.

A different approach is taken in this part of the work, starting with minimal data, simple models and a single activity of interest then iteratively working outwards from there, adding complexity where it will aid more consistent classification, measuring the results and then repeating the cycle.

6.2 SUMMARY OF ITERATIVE DEVELOPMENT CYCLE

During this part of the work eight iterations of goal directed development are undertaken and evaluated, starting with a simplistic evaluation of the best ESN model developed during the second, tool development phase that only looked at the data

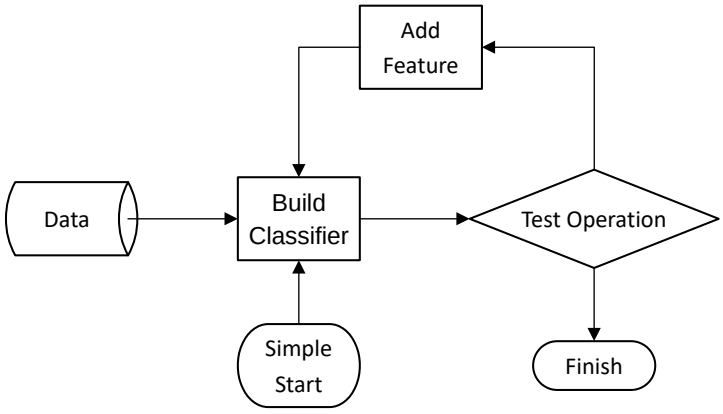


Figure 6.1: Iterative Development from Simple to Satisfactory

from the laboratory based, scripted activities; to the development of a ESN model based on earlier learnings and using the data collected during real–world, unscripted riding activities through to a satisfactory, best–so–far, ESN model that manages to correctly classify five out of the seven mounts that it is tested on. These iterations are summarised here and then expanded on in the following pages.

Section 6.4 on page 147 describes iteration 01–01, the first iteration in this series and in this first iteration using real–world, unscripted activity data the author decided to test the best performing ESN model from the phase two work against the real–world data in a somewhat simplistic attempt to see if the model developed from the scripted, laboratory based data had validity when used with real–world data.

Section 6.5 on page 149 describes iteration 02–01. This is the first ESN model built specifically for the data captured from the unscripted, real–world riding activities and is used as the base model against which to compare the following design iterations. This model incorporated much of the learning that had been gained from the phase two activities. Following the simplification design strategy, the process of producing, tuning and testing the ESN model was simplified and standardised. A Debian based, multi–core, virtual server, set up within the Amazon Elastic Cloud Compute environment, running R and RStudio is used for this and subsequent design iterations. This platform then became the standardised (and easily cloned) platform for running the PSO parameter search during the tuning cycle for each subsequent model.

Section 6.6 on page 160 describes iteration 03–01. The primary purpose for this design iteration is to look at a suggestion in Jaeger (2005, p. 44) to beware of symmetric input. According to Jaeger, when the input values are roughly symmetrical around zero then the ESN using a sigmoid actuator may not learn effectively. The Gyroscope signal is very much symmetrical around zero. The input signal used with this iteration will have a constant offset added to it so that it is no longer symmetrical around zero and is instead offset from zero.

Section 6.7 on page 169 describes iteration 04–01. The design goal for this iteration is to test a simple ensemble using both the gyroscope data as input plus the output from a much smaller, 60 neuron, ESN. The idea for this iteration came from comments by (Lukoševičius, 2012, p. 669) on the possibility of splitting a reservoir into different populations with differing parameters as a way of dealing with multiple time scales within the same input stream.

Section 6.8 on page 180 describes iteration 04–02. This iteration is similar to iteration 04–01 except that it builds and tests the ESN model using parameters from a earlier iteration. This parameter set then becomes the standard parameter set for the following design iterations.

Section 6.9 on page 190 describes a set of design iterations that use under-sampled training data. While Section 6.10 on page 198 describes a set of design iterations that use different features generated from the the Gyroscope data using a Butterworth filter.

Section 6.11 on page 206 describes a set of iterations (including iteration 07-01) that add Acceleration data to the Gyroscope data to see if this improves classification. As was noted during the second, tool development phase, the raw Accelerometer data as three separate axis are somewhat inconsistent and so for this iteration set the author uses the "net acceleration power" instead of the individual axis.

Section 6.12 on page 224 describes design iteration 08-01. This iteration is essentially an attempt to demonstrate the robustness of ESN classification of punctual activity by using an alternate R code library written by Mantas Lukoševičius, an ex–student of Jaeger's. Lukoševičius had posted some public code on the internet in various languages with a simple implementation of an ESN. A copy of the Lukoševičius "R" code is run against the data used in iteration 07-01.

6.3 REAL WORLD DATA

6.3.1 *Background*

When the real world data was gathered the intention was to record as much of a complete riding session as possible. Horses don't stand around in paddocks or barns with saddles on their backs and reins in their mouths waiting to be ridden. A horse that is kept in a paddock will spend most of its day moving around, eating and interacting with other horses in its herd. A horse kept in a barn will typically spend less time eating (as it is normally fed more concentrated food that is easier for people to handle) but will still spend time moving around and doing barn-type horse activities.

This means that the horse has to be prepared for riding, ridden and then there are post riding activities associated with putting the horse back into its paddock or barn. In many cases, the rider is responsible for preparing and putting away her own horse. These preparation and put-away phases are considered (by the author) to be part of the riding session and so, as far as possible, were recorded.

The significance of the presence or absence of preparation and post-riding activities becomes much more obvious when looking at the false positives from classification. During the Masters work when a possible pattern that seemed to be consistently

present during stirrup mounts was identified, it was theorised that a similar pattern might well be prevalent for certain types of backwards and forwards ”brushing” activity and such activity is typically encountered during preparation and post riding when the horse gets brushed to clean off dirt and sweat. The expectation going into this phase was that if false positives were encountered then they were most likely to occur during brushing.

6.3.2 *Classification Outcomes*

In a two class classification problem such as that within this part of the work, there are four possible outcomes. A data point within a Mount sequence is correctly classified as a Mount (**True Positive**) or incorrectly classified as a Base Class (**False Negative**). A data point within a Base class sequence is correctly classified as a Base class instance (**True Negative**) or incorrectly classified as a Mount (**False Positive**).

Within this work, a Mount is intended to be used as a trigger event and so if, for example, a mount data sequence contained (say) eight data points at 10hz then successful classification of that mount is achieved if one or more of the eight data points produce an output from the ESN model that is high enough to be distinguished as a Mount. Within the earlier phases of this work we saw that the ESN output response for a Mount was usually an inverted V with the ESN output quickly rising from around zero to a peak of some sort and then quickly falling back to around to zero. Given this scenario of only needing one of the eight Mount data points to be classified correctly when evaluating one ESN model against another there are two key outcomes that are of interest and they are True Positives (one or more correct classifications within a Mount sequence) and False Positives (one or more contiguous sequences of Mount classifications during a Base class sequence). We are also somewhat interested in True Negatives but much less so as the number of Base class data points is far greater and tends to overwhelm the comparison between different models. However, we are not at all interested in the number of False Negatives when comparing models.

To reiterate, given a Mount sequence of (say) eight data points where one of the data points is correctly classified as a Mount (True Positive) and seven are incorrectly classified as Base class (False Negative) versus a situation where three data points are correctly classified and five are incorrectly classified then within this work, both situations are considered equally correct and desirable. As a result, False negatives are reported in the following confusion tables but not commented on.

6.3.3 *Data Description*

The data files that were used during this phase are listed in the table 6.1, along with information on how they were used (training, tuning or testing) and if they encapsulate preparation or post-riding activities.

Files described as “Train-Tune” were used to train the ESN and tune the parameter search process. Files described as “Test” were held back and only used to test an optimised set of parameters that had been trained in an earlier process. There were

Session	Participant	Usage	Preparation	Post Riding
0716-2	RA	Train-Tune	Yes	No
0719-1	RB	Test	No	Yes
0812-2	RA	Test	Yes	No
0829-1	RC	Test	No	No
0830-3	RD	Test	Yes	Yes
0902-1	RE	Test	Yes	Yes
0902-2	RC	Test	Yes	Yes
0912-1	RF	Train-Tune	Yes	Yes
0912-2	RG	Train-Tune	Yes	No
0913-3	RF	Train-Tune	Yes	Yes

Table 6.1: Real world data capture sessions

ten real world data sets available after, four data sets were used for training and tuning and six were held back for testing. Five data sets included both preparation and post-riding phases. One data set contained two mounts and dismounts as the participant dismounted and then re-mounted during their riding session.

Unfortunately, there are not three usable real-world data sets from any one participant and the author was keen to have at least one collection of three so that intra-participant training and testing might be tested. To resolve this two lots of two mounts from the LA Laboratory data set 0813-1 LA were included and used during training and tuning and the 0719-1 LA data set was reserved for testing. Other data sets were allocated to "Train-Tune" or "Test" in a somewhat arbitrary manner that resulted in a relatively balanced mix. In the end, there was insufficient time and data to test intra-participant classification.

6.4 LABORATORY ESN MODEL WITH REAL WORLD DATA (01-01)

It had been assumed that the ESN models that were built based purely on the laboratory data captured during the scripted sessions would not work with real world data because of artefacts within the laboratory data caused by following the script. However, it was decided to test this rather than just assuming that the outcome would be unsatisfactory.

6.4.1 *Description (01-01)*

Training Data: Subset of 0813-1 LA (laboratory)
Testing Data: Full 0719-1 RB
Included Sensors: Accelerometer and Gyroscope
The sensor readings are normalised, real-values ranging between -1 and 1
No outliers were removed.

Table 6.2: Run Time Parameters for ESN iteration 01-01.

Name	Value	Justification
Input values	6 vectors of data	Input choice
Classes	Mount, Dismount	Prior comparison
Number of neurons	999	Search result
Input scaling	0.6922	Search result
Neuron leak rate	0.7367	Search result
Regularisation	4.5639	Search result
Spectral Radius	1.0000	Search result

6.4.2 *Results and Discussion (01-01)*

At first look of figure 6.2 the assumptions are confirmed and clearly this ESN model is not a suitable candidate classifier for Mounts and Dismounts as false positives far outnumber the true positives. However, that this model did correctly classify the Mount and Dismount as shown by the arrows on the figure and confirmed in Figure 6.3 (zoomed in on the Mount) gives some comfort that additional work and more realistic training data may well produce the results that are desired.

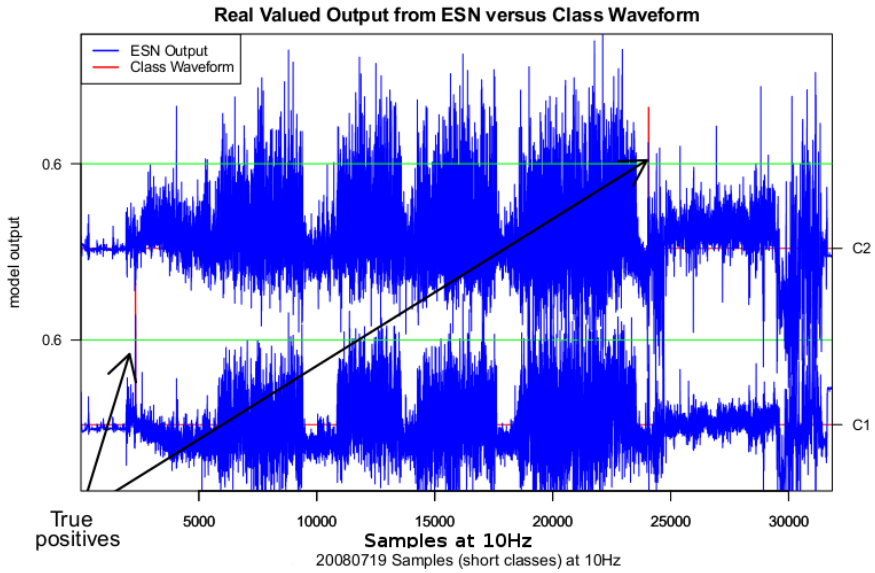


Figure 6.2: Testing Real World Data against Laboratory Data Model

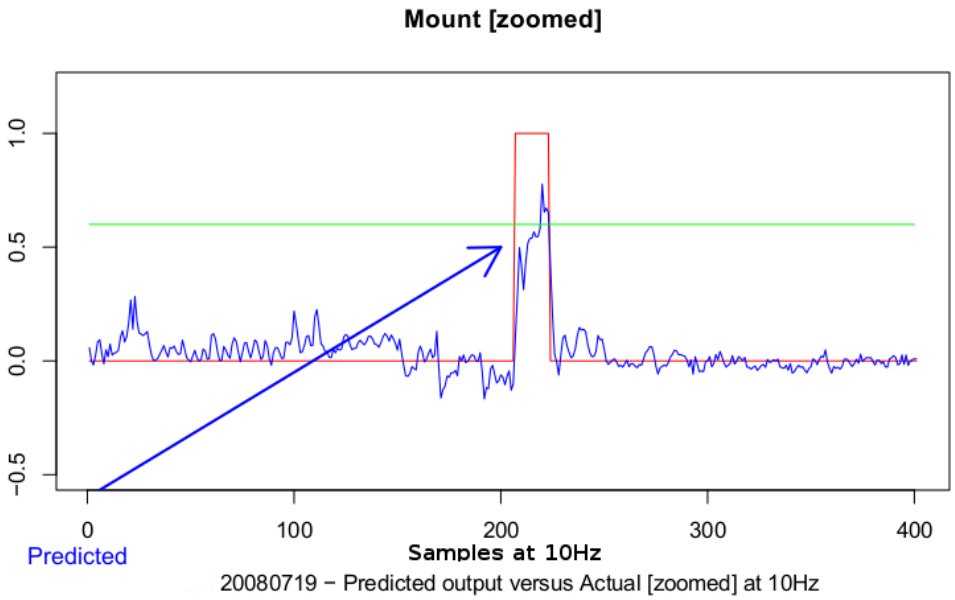


Figure 6.3: Zoomed in Test of Real World data using Laboratory Model

6.5 INITIAL - ESN MODEL WITH REAL WORLD DATA (02-01)

The aim of this process was to build an initial, base ESN model based on what had been learned during phase one and two, using real world data. The goal in this first of the iterative steps was to build a simple model that works to some extent that will then be the base for adding features that improve its ability to classify. Future design iterations will be tested against this base when features (and complexity) are added to see if the additional features add to classification accuracy and if they do, at what cost.

6.5.1 *Problem Identification (02-01)*

Classifying real world activity is generally considered much more difficult than classifying scripted activity. Activity classification researchers such as Foerster et al. (1999), Bao and Intille (2004) and Ravi et al. (2005) have described achieving high classification levels for scripted activities recorded within laboratory situations but noted that their classification levels fell dramatically (24%–66.7% drop) when used with real-world, unscripted activities. This design iteration is the first within this work that demonstrates the ability of an ESN to classify punctual activity where the ESN model parameters have been optimised for real-world data. It is an important first step in this phase of the research as it sets the benchmark for the design iterations that follow.

6.5.2 *Motivation (02-01)*

The envisaged wearable coach that is the end point for this research is designed to be used by riders as they do their everyday training in the real world of horse riding. As a result it is crucial that any classifier that is being considered for use within this coaching system be capable of classifying unscripted activities across a variety of riders in a variety of real situations.

6.5.3 *Design Goal (02-01)*

The Design Goal for this iteration was to generate a base ESN model that was capable of successfully classifying mounts from the set of test data where the ESN model was simple and based on the experience gained to-date.

6.5.4 *Measure of Success (02-01)*

The measure of success was to consider a class as being correctly classified if the model output was greater than 0.6 during some point within the Mount sequence.

6.5.5 *Data Description (02-01)*

Training Data; concatenation of:

- Partial 0813-1 LA (laboratory) – Subset from end including two mounts
- Partial 0716-2 RA – Central (riding) region reduced
- Partial 0912-1 RF – Central (riding) region reduced

Tuning Data; concatenation of:

- Partial 0813-1 LA (laboratory) – Subset from beginning including two mounts
- Partial 0912-2 RG – Central (riding) region reduced
- Partial 0913-3 RF – Central (riding) region reduced

A portion of data between the Mount and Dismount, recorded while the participant was riding the horse, was cut from Training and Tuning data to assist in balancing class and non-class sizes.

Testing Data; concatenation of:

- Full 0719-1 RB
- Full 0812-2 RA
- Full 0829-1 RC
- Full 0830-3 RD
- Full 0902-1 RE

- Full 0902-2 RC

No part of the testing data-sets were removed. That is all datasets in their entirety were concatenated together and used for testing.

6.5.6 *Static Parameter Description (02-01)*

The following static parameters were used during this iteration:

- Error Cost Function: (conditional mean square error class + conditional mean square error non-class)$/2$
- Included Sensors: Gyroscope only
- The sensor readings are linearly transformed, real-values ranging between -1 and 1. No outliers were removed
- Offset added to sensor data: Null

6.5.7 *PSO Parameter Description (02-01)*

PSO parameter ranges for search			
Name	Id	Range	Justification
Regularisation	α	0.0001 to 5	S. 5.4.15
Number of Neurons	N_x	100 to 360	Lukoševičius (2012) NB.
Input Scaling		0.0001 to 1	Full range
Leaking Rate	a	0.0001 to 1	Full range
Spectral Radius	$\rho(\mathbf{W})$	0.6000 to 1	S. 5.4.15 NB.
ESN Repeats		7	S. 5.4.15
Particles	S	14	Bendtsen. (2012)
Maximum Generations		25	S. 5.4.15 NB.

Table 6.3: PSO Parameter Ranges for Iteration 02-01

A PSO search was run using the parameter ranges listed in table 6.3 to find a good set of meta–parameters to use to train the ESN model for classification. The PSO search parameters are in two parts, the first part includes the ESN meta-parameter ranges to search across and include Regularisation, Number of Neurons, Input Scaling, Leaking Rate and Spectral Radius. The chosen ranges were selected based on the work done within the Exploration phase of this work, Section 5.4 and, in particular, 5.4.15. Input Scaling and Leaking Rate cover the full range for both of these parameters with a 0.0001 start point. Please note that the lower bound for the number of neurons has been chosen based on both a recommendation from Jaeger (2005) and prior work while the upper bound for the number of neurons was chosen both

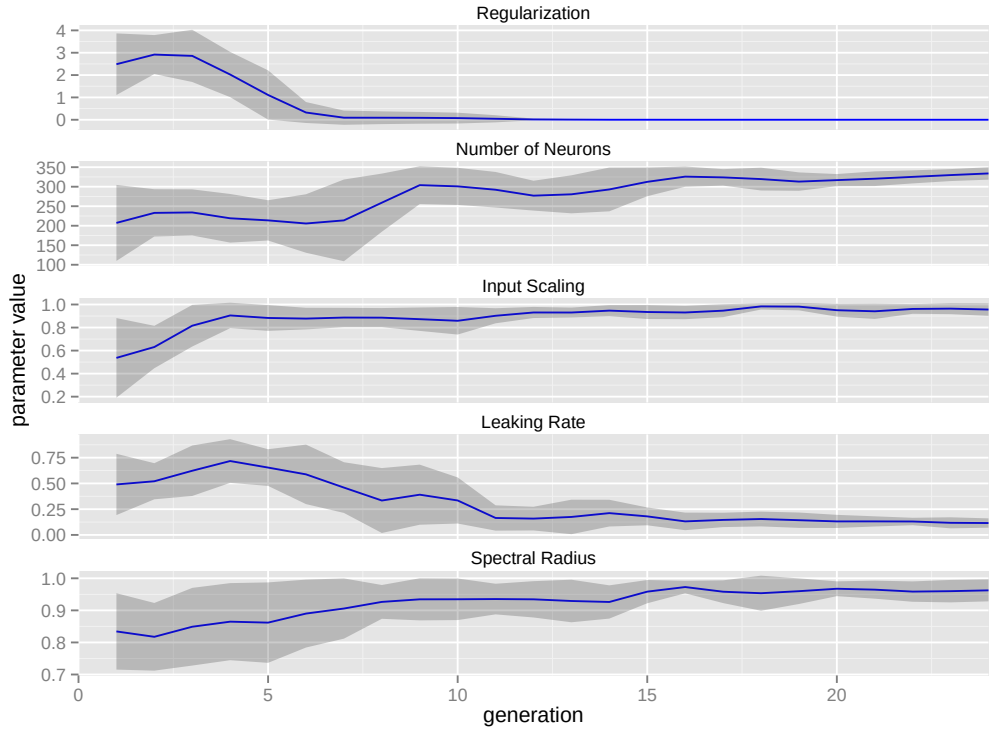


Figure 6.4: Parameter Optimisation for 02-01 Model

on advice from Lukoševičius (2012) that model parameters can be tuned on smaller models and then scaled up later and pragmatically based on the earlier work with reasonable results being found within this range while managing the additional, exponential growth in computer processing that comes with higher neuron numbers. In addition, the earlier work demonstrated that a lower bound of 0.6 for Spectral Radius is a good starting point based both on processing time practicalities and a recommendation from Lukoševičius that Spectral Radius rates above 0.7 tended to give better results.

The second set of parameters, ESN Repeats, Particles and Maximum Generations are used by the PSO search control the search itself and again the parameter values were chosen based on the earlier exploratory work. The initial value of the internal ESN inter–neuron weights are set stochastically and so any new ESN model that is generated, even though it has the same meta–parameters will produce a slightly different result because of the differing inter–neuronal weights. To partially account for this the PSO search code recreates (repeats) each model that it tests and then uses the average Error Cost function across all of the repeats. For this work 7 repeats was chosen versus the 3 repeats that was typically used within the exploratory phase work. The additional repeats is a somewhat arbitrary balance between the additional computing resources needed to create, train and test each iteration versus the additional confidence that can be drawn from averaging a higher number of instances. The move to running the PSO search on a multiprocessor Debian virtual machine in the Amazon Compute Cloud enables the repeats to be done in parallel on separate

processors and so the overall elapsed search time was controlled at the expense of more processors. The number of particles was set heuristically based on a combination of the recommendation that accompanies the R PSO package that was used to conduct the PSO search (see (Bendtsen., 2012, p.7)) and experience running PSO searches during the exploration phase. Bendtsen's manual for the use of the PSO package suggests that a useful starting point for the number of particles is related to the number of dimensions that the search is conducted across (in this work 5 dimensions) and is calculated as $floor(10 + 2 * sqrt(length(par)))$ or 14 and then adjust that number slightly to a number that suits the problem at hand. The experience from the exploratory work was that a particle count of 14 gave useful results while consuming reasonable compute resources and so that number of particles has been used throughout. The final parameter used by the PSO search is a stopping condition and these series of searches has used a maximum number of generations (set to 25) as the experience from the exploratory work was that was sufficient generations to settle on a good–enough result.

This search ran on an Amazon ECC virtual 8 processor machine running Debian. The ESN parameters with the lowest fitness score are listed in table 6.4. During parameter optimisation the Regularization parameter quickly dropped to its minimum value. Input Scaling took slightly longer (about 15 generations) to drift towards its maximum value. Over a similar number of generations, Leaking Rate stabilised at around 0.13 and Spectral Radius floated around 0.9. Number of Neurons took around 17 generations to stabilise between 300 and 360.

6.5.8 *Run time Parameter Description (02-01)*

Once the optimised ESN model meta–parameters were found from the PSO search they were then used as run time parameters for an new ESN model that was built, trained and tested on a Windows 7 based PC running an i7 processor and 8GB RAM. This ESN model was then trained with the previously unseen riding real–world concatenated dataset using the parameters listed in table 6.4.

Table 6.4: Run Time Parameters for ESN iteration 02-01.

Name	Value	Justification
Input values	3 vectors of data	Chosen input
Classes	Mount only	Chosen class
Number of neurons	336	Search result
Input scaling	0.9252	Search result
Neuron leak rate	0.1379	Search result
Regularisation	0.0001	Search result
Spectral Radius	0.9455	Search result

6.5.9 *Training Results (02-01)*

Training completed in 7.71 seconds elapsed.

Table 6.5: Computing times for ESN iteration 02-01 Training.

Descriptor	Training Time in Seconds
User Time	7.28
System Time	0.09
Elapsed Time	7.50

6.5.10 *Test Results (02-01)*

The model was trained once using the parameters found during the parameter optimisation process and with the training data specified, with resource usage and results as listed in table 6.5 and then tested against a dataset that consisted of a concatenation of each of the test files. The results of testing against the ESN model with the concatenated file are shown in Table 6.8. The Classification rate is calculated by dividing the number of Classified Mounts by the total number of Mounts in the concatenated test file (7). A Mount is deemed classified if the ESN output is 0.6 or greater for at least one sample during the Mount segment.

CONCATENATED

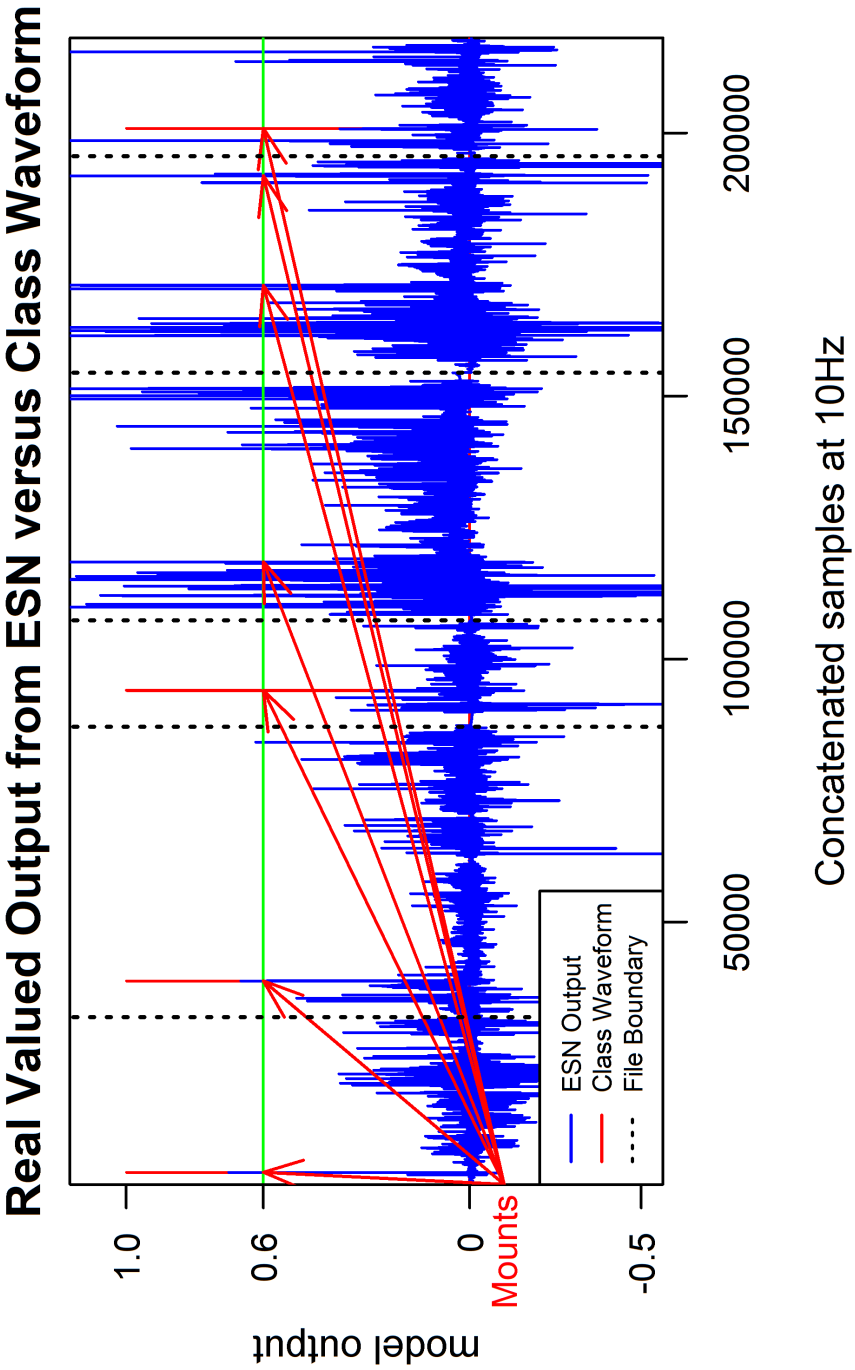
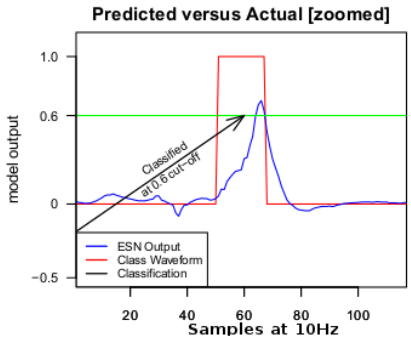
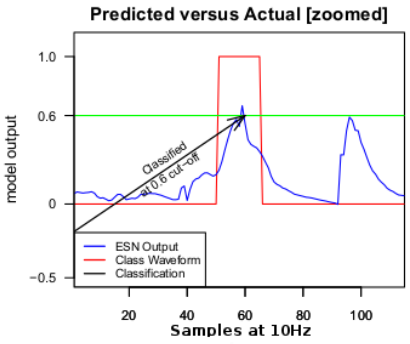


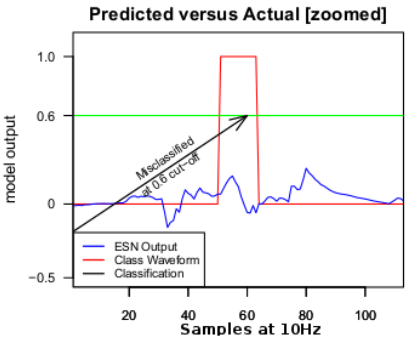
Figure 6.5: Concatenated ESN Output (02-01)



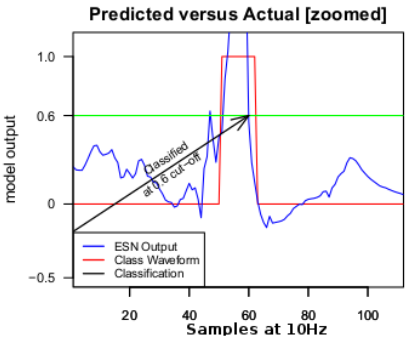
(a) 0719-1 RB #1



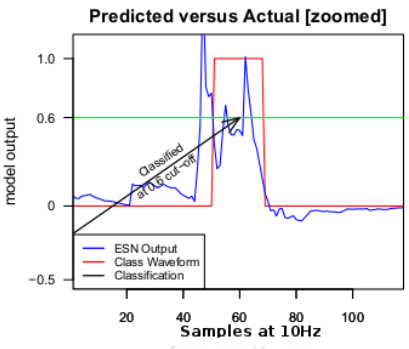
(b) 0812-2 RA #1



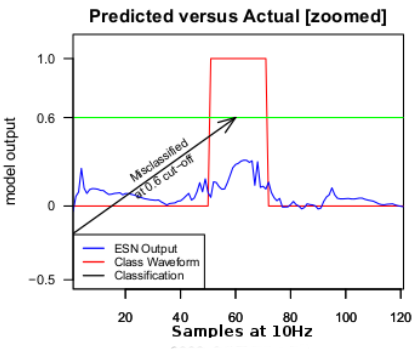
(c) 0829-2 RC #1



(d) 0830-3 RD #1

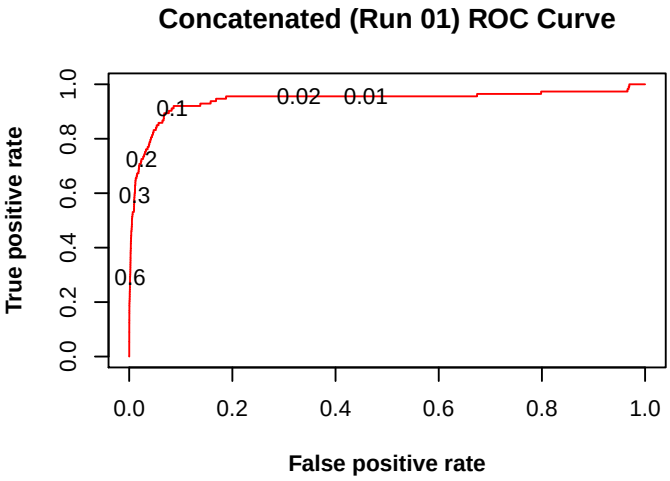


(e) 0902-1 RE #2

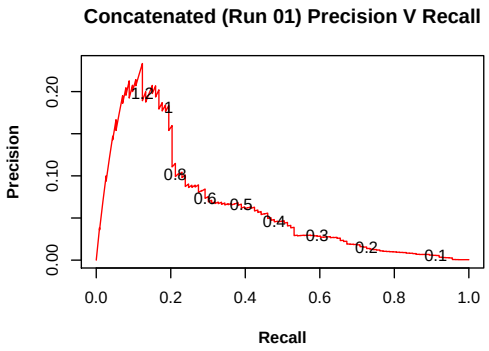


(f) 0902-2 RC #1

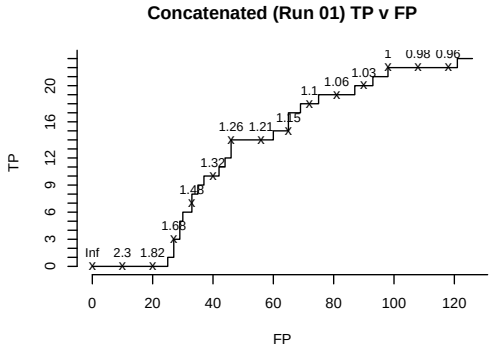
Figure 6.6: Concatenated Mounts (02-01); 0902-1 RE #1 not shown



(a) ROC Curve



(b) Precision V Recall



(c) TP V FP

Figure 6.7: Concatenated datasets performance plots (02-01)

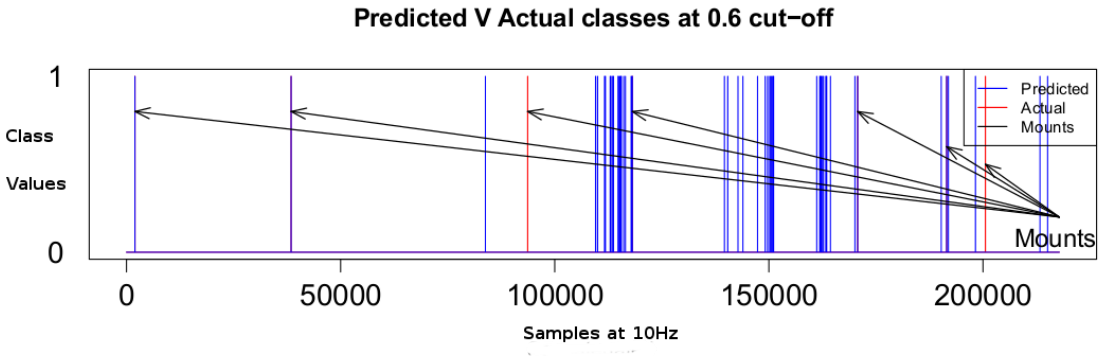


Figure 6.8: Concatenated datasets Classes (02-01)

Table 6.6: Processing Times for Concatenated Datasets (02-01) Test.

Description	Time in Seconds
User Time	29.94
System Time	00.89
Elapsed Time	30.83

Table 6.7: Classification Metrics for Concatenated Datasets (02-01) Test.

Metric	Value
ESN Test Error	0.2046
RMSE	0.0874
AUC	0.9418

Table 6.8: Testing Results for Concatenated Datasets (02-01).

Classification measures at 0.6 Cut-Off		
Classification rate	0.714	
	Base Class	Mount Class
Precision	0.9996	0.0737
Recall	0.9981	0.2920
Confusion Matrix at 0.6 Cut-Off		
	Predicted	
Actual	Base Class	Mount Class
Base Class	217295	415
Mount Class	80	33

A very satisfactory result for the initial iteration, based on the area under the ROC curve of 0.9418 (Table 6.7). However, high True Positive rates (0.7 and above, see Figure 6.7a) are only achieved at low response levels from the ESN model at around 0.2. This is much lower than the selected 0.6 cut-off level. As the cut-off level is raised from 0.2 towards 0.6, the Precision rate rises slowly while the Recall rate drops quickly (Figure 6.7b). Looking at the number of True Positives graphed against the number of False Positives (Figure 6.7c), it can be seen that the cut-off level would need to be raised well above 1.0 simply to get the total number of False Positives below 100 with this particular ESN model. Most of the False Positives are associated with two participant sessions, 0830-3 RD and 0902-1 RE and the False Positives mostly occur prior to mounting during these two sessions. Both of these sessions included extensive riding preparation activities including brushing the horse, see Figure 6.5 and Figure 6.8.

Five of the seven mount sequences are associated with an ESN response of 0.6 or greater, giving a classification rate of 0.714 at this cut-off level. Two mounts from two participants (Figures 6.6a & b) are associated with a response that just barely reaches 0.6, three mounts from another two participants (Figures 6.6d & e) are associated

with very strong ESN responses while the last two mounts from a single participant (Figures 6.6c & f) are associated with a noticeable but weak response from the ESN model.

6.5.11 *Discussion (02-01)*

The results obtained represent the base which can be progressively improved by adding techniques learned during the second, exploration phase. Improvement may be difficult considering the high area under the ROC curve of 0.9418 overall and with five of the six sessions achieving individual area under the ROC curve figures of 0.98 and 0.99 and above.

The challenge now is to improve the results, particularly the unsatisfactory results from the 0829-2 RC session while maintaining or improving the other results. This will be especially challenging because the output from the ESN model is not homogeneous across participant sessions. The two RC data-sets have a response from the ESN model that favours successful classification at around the 0.2 cut-off level, RE and RD produced much higher responses that favour successful classification at around the 1.0 cut-off level while RB and RA produced more moderate responses that favour successful classification just above 0.6.

In addition, the large skew is class distribution makes measuring the ESN performance a challenge. The intention with measurement is to favour the area under the ROC curve as the primary measure as it is relatively immune to class skew. In particular, the overall area under the curve for the full concatenated test file will be used rather than individual session areas under the curve. The secondary measure will be the classification rate, as defined in section 6.5.10.

6.5.12 *Results versus Goals (02-01)*

The goal has been exceeded, based on the measure, as four of the six datasets resulted in an ESN response of greater than 0.6 at some point during the Mount sequences for the concatenated test dataset.

6.6 OFFSET - ESN MODEL WITH REAL WORLD DATA (03-01)

According to Jaeger (2005, p. 44), when the input values are roughly symmetrical around zero then the ESN using a sigmoid actuator may not learn effectively. The Gyroscope measures rotation acceleration and as a result the signal from the gyroscope is very much symmetrical around zero. This symmetry can be seen in Figure 4.13 on page 88, although this figure shows the data prior to normalisation. The normalisation process centres the Gyroscope data on zero.

This design iterations builds an ESN using data which has been offset from its original values, ensuring that it is no longer centred on zero. The results from this iteration will then be compared against the initial base ESN model.

In order to provide the most useful measures the same data was used to train and test the new model as that used for the initial model, excepting, of course, that the data used in these iterations had a positive offset added to it. An arbitrary offset of $+0.45$ is used in this iteration, no alternate offset values are tried.

6.6.1 *Problem Identification (03-01)*

The initial design iteration that used real world data produced good results but did not meet the overall goal of this research as the number of False Positives were too high. The problem now is to find a way to maintain or improve the classification rate while increasing the ability of the classification engine to discriminate the activity of interest.

6.6.2 *Motivation (03-01)*

The motivation for this design iteration is a simple one and is to further improve the classification ability by altering the characteristics of the input data.

6.6.3 *Design Goal (03-01)*

The goal in this iterative step was to demonstrate an improvement in class discrimination compared with the initial model based on the use of an offset for the input data.

6.6.4 *Measure of Success (03-01)*

An overall improvement in the area under the ROC curve while maintaining or improving the classification rate as defined in section 6.5.10.

6.6.5 *Data Description (03-01)*

Training Data; concatenation of:

- Partial 0813-1 LA (laboratory) – Subset from end including two mounts

- Partial 0716-2 RA – Central (riding) region reduced
- Partial 0912-1 RF – Central (riding) region reduced

Tuning Data; concatenation of:

- Partial 0813-1 LA (laboratory) – Subset from beginning including two mounts
- Partial 0912-2 RG – Central (riding) region reduced
- Partial 0913-3 RF – Central (riding) region reduced

Data from the riding phase was removed exactly as done for the initial model. In addition 0.45 was added to all sensor reading in order to offset the data so that it is no longer centred on zero.

Testing Data; a concatenation of:

- Full 0719-1 RB
- Full 0812-2 RA
- Full 0829-1 RC
- Full 0830-3 RD
- Full 0902-1 RE
- Full 0902-2 RC

The entire data file was always used for testing files. The same 0.45 offset is added to the test files.

6.6.6 *Static Parameter Description (03-01)*

The following static parameters were used during this iteration:

- Error Cost Function: (conditional mean square error class + conditional mean square error non-class)$/2$
- Included Sensors: Gyroscope only
- The sensor readings are linearly transformed, real-values ranging between -1 and 1. No outliers were removed
- Offset added to sensor data: $+0.45$

6.6.7 *PSO Parameter Description (03-01)*

A PSO search was run using the parameter ranges listed in table 6.9. This search ran on an Amazon ECC virtual 8 processor machine running Debian. The ESN parameters with the lowest fitness score are listed in table 6.10.

During parameter optimisation the Regularization parameter quickly dropped to its minimum value. Input Scaling took slightly longer (about 15 generations) to drift towards its maximum value. Over a similar number of generations, Leaking Rate

PSO parameters range for search			
Name	Id	Value	Justification
Regularisation	α	0.0001 to 5	Prior S. 6.5.7
Number of Neurons	N_x	100 to 360	Prior S. 6.5.7
Input Scaling		0.0001 to 1	Prior S. 6.5.7
Leaking Rate	a	0.0001 to 1	Prior S. 6.5.7
Spectral Radius	$\rho(\mathbf{W})$	0.6000 to 1	Prior S. 6.5.7
ESN Repeats		7	Prior S. 6.5.7
Particles	S	14	Prior S. 6.5.7
Maximum Generations		25	Prior S. 6.5.7

Table 6.9: PSO Parameter Ranges for Iteration 03-01

Figure 6.9: Parameter Optimisation for 03-01 Model

stabilised at around 0.13 and Spectral Radius floated around 0.8. Number of Neurons took around 17 generations to stabilise between 300 and 360. This was very similar to what happened with the initial model except that all parameters except Regularisation settled to lower values, reflecting the impact of the higher valued input (once the offset is added). The largest difference was Spectral Radius, with this settling at 0.75 versus 0.94 for the initial model.

6.6.8 *Run time Parameter Description (03-01)*

The parameters that produced the lowest error cost during optimisation are shown in table 6.10 and the ESN model that was built for this set of design iterations was built, trained and tested on a Windows 7 based PC running an i7 processor and 8GB RAM with these parameters. The initial parameters are also listed for comparative purposes.

Table 6.10: Run Time Parameters for ESN iteration 03-01.

Description	**Current Iteration**	Justification	Initial Iteration
Input values	3 **vectors of data**	Input choice	3 vectors of data
Classes	**Mount only**	Class choice	Mount only
Number of neurons	**332**	Search	336
Input scaling	**0.8486**	Search	0.9252
Neuron leak rate	**0.1283**	Search	0.1379
Regularisation	**0.0001**	Search	0.0001
Spectral Radius	**0.7589**	Search	0.9455

6.6.9 *Training Results (03-01)*

The model was trained once using the best parameters found during the parameter optimisation process and with the training data specified, with resource usage and results as listed in table 6.11, with comparative figures for the initial model. Training completed in 7.5 seconds elapsed.

Table 6.11: Training Results for ESN iteration 03-01.

Description	**Current Iteration**	Initial Iteration
User Time in seconds	**7.55**	7.28
System Time in seconds	**0.09**	0.09
Elapsed Time in seconds	**7.71**	7.50

6.6.10 *Test Results (03-01)*

The model was trained once using the parameters found during the parameter optimisation process and with the training data specified, with resource usage and results as listed in table 6.11 and then tested against a dataset that consisted of a concatenation of each of the test files, as in iteration 02-01. The results of testing against the ESN model with the concatenated file are shown in Table 6.14. A Mount is deemed classified if the ESN output is 0.6 or greater for at least one sample during the Mount segment. Classification rate is as defined in section 6.5.10.

Table 6.12: Processing Times for Concatenated Datasets (03-01) Test.

Iteration Id	**03-01**	02-01
User Time in seconds	29.12	29.94
System Time in seconds	00.64	00.89
Elapsed Time in seconds	29.76	30.83

Table 6.13: Classification Metrics for Concatenated Datasets (03-01) Test.

Iteration Id	**03-01**	02-01
ESN Test Error	0.2719	0.2046
RMSE	0.0945	0.0874
AUC	0.8697	0.9418

Table 6.14: Results for Concatenated Datasets iteration 03-01.

Measures at 0.6 Cut-Off				
Description	**03-01**		02-01	
Classification rate	0.571		0.714	
	Base Cl	**Mount Cl**	Base Cl	Mount Cl
Precision	0.9996	0.0414	0.9996	0.0737
Recall	0.9969	0.2566	0.9981	0.2920
Confusion Matrix at 0.6 Cut-Off				
	Predicted 03-01		Predicted 02-01	
Actual	**Base Cl**	**Mount Cl**	Base Cl	Mount Cl
Base Cl	217039	671	217295	415
Mount Cl	84	29	80	33

Both the area under the ROC curve and the classification rate were worse for this Offset iteration than for the initial iteration. At 0.8697, (Table 6.14) the area under the ROC curve is much smaller than the initial 02-01 iteration. In addition the classification rate at the 0.6 cut-off has dropped to 0.571 from 0.714 with only four of the seven mount sequences being associated with ESN output response of more than 0.6.

Comparing Figure 6.10 (03-01) with Figure 6.5 (02-01), it can be seen that one effect of adding the 0.45 offset has been to push the output response to the Base class slightly higher, in general (mean increased from 0.0203 to 0.0266; standard deviations from 0.0838 to 0.0898), while the response to the Mount class is similar or less (mean decreased from 0.5239 to 0.3869, standard deviation from 0.4561 to 0.4062). Also see sub-figures within Figure 6.11, and in particular Figure 6.11e. These shifts are demonstrated via histograms in Figure 6.12. Four of the seven mount sequences are associated with an ESN response of 0.6 or greater, giving a classification rate of 0.571 at this cut-off level. This is less than the classification rate achieved for the initial, 02-01 iteration. Two mounts from two participants (Figures 6.11a & b) are again associated with a response that just barely reaches 0.6, two mounts from another two

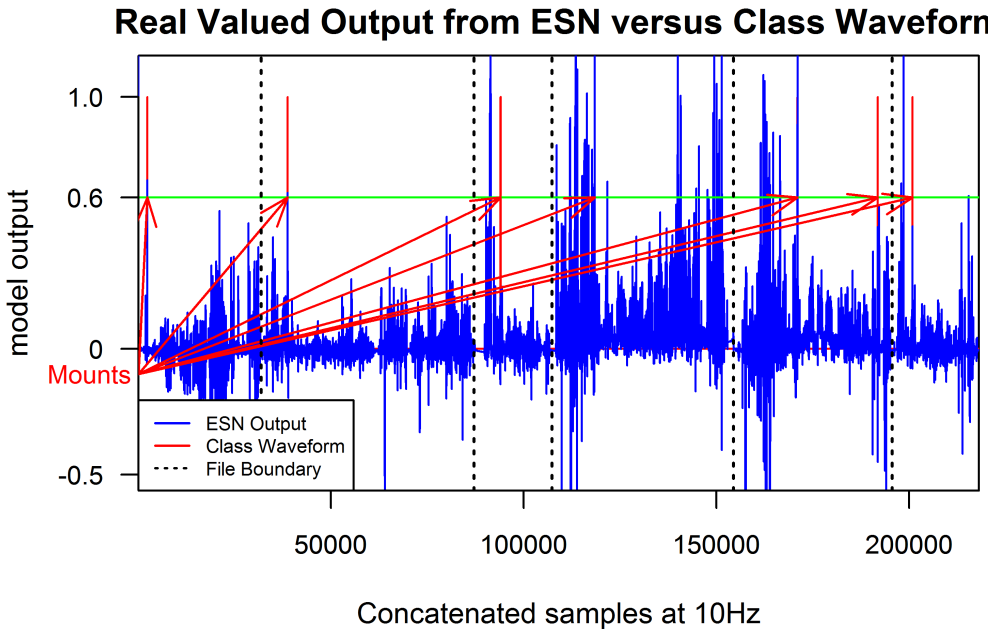
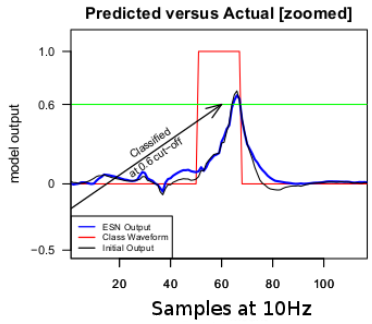


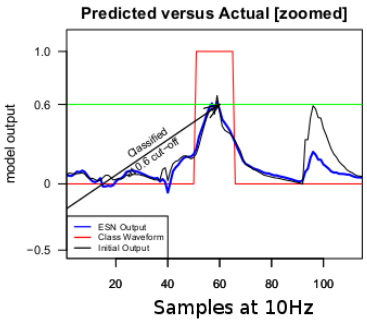
Figure 6.10: 03-01 - ESN Output for Concatenated Test Data-sets

participants (Figure 6.11d) are associated with very strong ESN responses while the last three mounts from two participants (Figures 6.11c, e & f) are associated with a weak response from the ESN model.

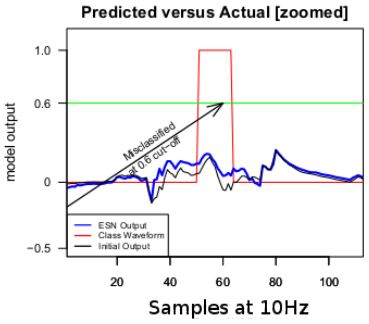
Figure 6.13a shows the lower ROC curve for this iteration compared with the initial 02-01 iteration. The current iteration demonstrates less discriminate power at lower response levels but similar discriminative power at the higher levels. The current iteration produces more False Positives at the selected 0.6 cut-off level, see Table 6.14 and Figure 6.14.



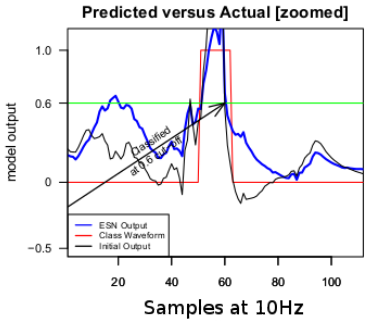
(a) 0719-1 RB #1



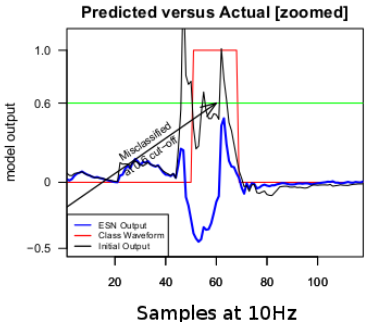
(b) 0812-2 RA #1



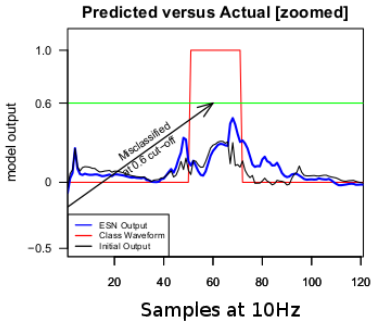
(c) 0829-2 RC #1



(d) 0830-3 RD #1

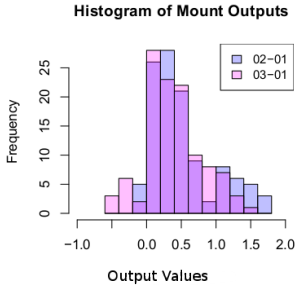


(e) 0902-1 RG #2

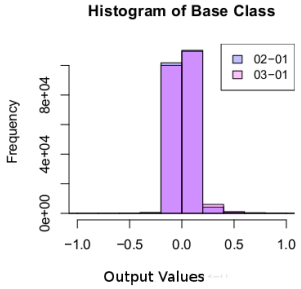


(f) 0902-2 RC #1

Figure 6.11: Mounts for Concatenated (03-01); 0902-1 RG #1 not shown

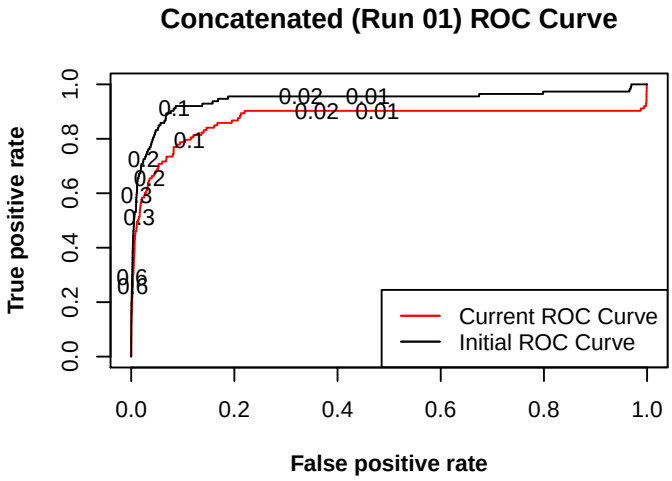


(a) Mount Class

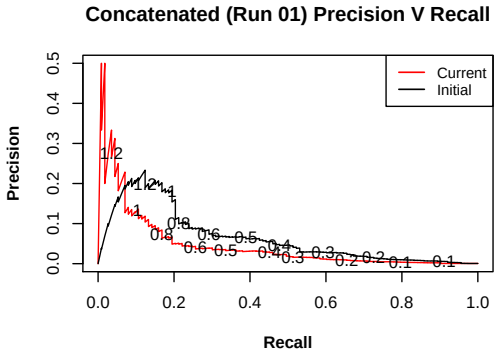


(b) Base Class

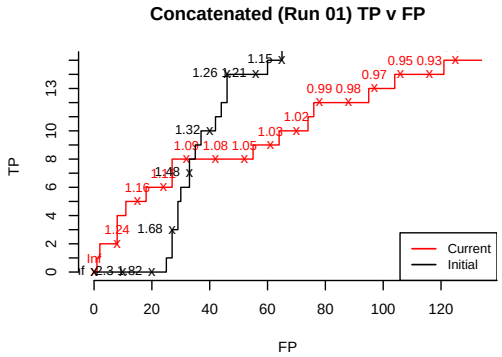
Figure 6.12: Comparing ESN Outputs 03-01 and 02-01



(a) ROC Curve



(b) Precision V Recall



(c) TP V FP

Figure 6.13: 03-01 - Performance Plots for Concatenated Data

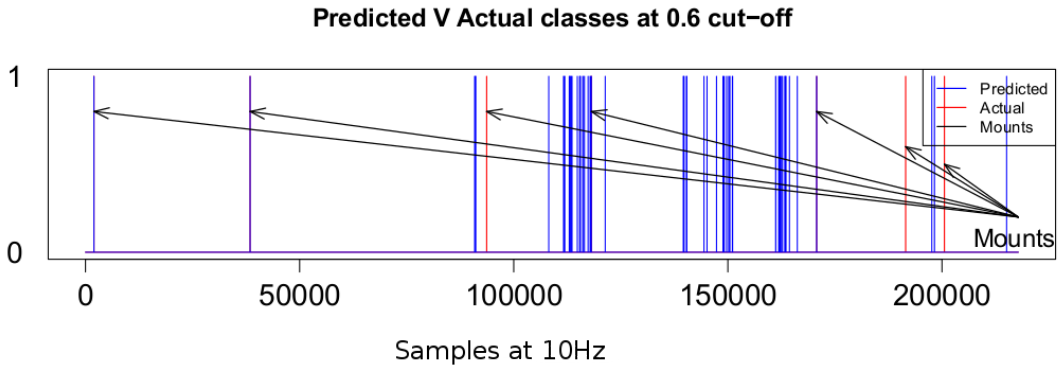


Figure 6.14: 03-01 - Classes for Concatenated Test Data

6.6.11 *Results versus Goals (03-01)*

The goals have not been met as the Area Under the Curve (AUC) and the classification rate at 0.6 have both fallen. The AUC fell from 0.9418 to 0.8697 while the classification rate fell from 0.714 to 0.571.

6.6.12 *Discussion (03-01)*

While the overall AUC fell, there were minor falls for 0719-1 RB (0.9923 to 0.9922) and 0902-2 RC (0.9870 to 0.9846); minor increases for 0812-2 RA (0.9994 to 0.9998) and 0830-3 RD (0.9828 to 0.9947); a major increase for 0829-2 RC (0.6630 to 0.9586) and a major decrease for 0902-1 RE (0.9967 to 0.6719). Thus there seems to be some variability between different participant sessions and while area under the ROC curve is a measure that is relatively insensitive to class bias it is not insensitive to model bias.

0902-1 RE, the participant session with the largest decrease in AUC contains 41,000 samples while 0829-2 RC, the participant session with the largest increase in performance has only 20,000 samples and so the overall results are probably more influenced by 0902-1 RE than by 0829-2 RC. At this stage in the work, overall AUC is the best performance measure and using it as the measure it must be concluded that this iteration did not improve the classification performance for the ESN model.

In addition, the selected cut-off level of 0.6 may not be optimal overall and in particular is not optimal for each participant session. This level will be maintained as the cut-off though, at least for the time being so that a consistent approach can be maintained throughout this phase of the work.

6.7 ENSEMBLE #1 - ESN MODEL WITH REAL WORLD DATA (04-01)

A number of authors including Chawla (2005), Z. Zhou and Liu (2006), Mollineda, Alejo, and Sotoca (2007) and X. Liu, Wu, and Zhou (2009) have suggested using ensembles of classifiers, especially with imbalanced classes and so this design iteration acknowledges the learning relating to the usefulness of small ESN classifier discussed in Section 6.1 and looks at a simple ensemble that uses the output of a smaller ESN classifier as additional input into the main ESN classifier. The aim of this set of design iterations is to build an ESN classifier using data which includes the output from a small, parameter optimised ESN in **addition** to the linearly transformed Gyroscope data from prior iterations. The goal in this iterative step was again to build and test a model using modified data and then measure that model against the initial model. In order to provide the most useful measures the same data was used to train and test the new model, excepting, of course, that the data used in these iterations includes an **additional signal**.

6.7.1 *Problem Identification (04-01)*

The initial design iteration that used real world data produced good results but did not meet the overall goal of this research as the number of False Positives were too high. The problem now is to find a way to maintain or improve the classification rate while increasing the ability of the classification engine to discriminate the activity of interest.

6.7.2 *Motivation (04-01)*

The motivation for this design iteration is a simple one and is to further improve the classification ability by altering the characteristics of the ESN model and its relationship with the input data.

6.7.3 *Design Goal (04-01)*

The design goal for this iteration is to test a simple ensemble using both the gyroscope data as input plus the output from a much smaller, 60 neuron, ESN. The idea for this iteration came from comments by (Lukoševičius, 2012, p. 669) on the possibility of splitting a reservoir into different populations with differing parameters as a way of dealing with multiple time scales within the same input stream. A decision was made to implement this idea by including the output from a much smaller model, rather than reprogramming the ESN library code to enable the splitting of the main reservoir into two or more populations with differing parameters. This alternate approach to including differing time scales is consistent with (Lukoševičius, 2012, p. 675) and other authors such as Jaeger and Haas (2004) who report that ensembles of ESN can sometimes dramatically improve performance over a single ESN.

6.7.4 *Measure of Success (04-01)*

An overall improvement in the area under the ROC curve while maintaining or improving the classification rate as defined in section 6.5.10.

6.7.5 *Data Description (04-01)*

Training Data; concatenation of:

- Partial 0813-1 LA (laboratory) – Subset from end including two mounts
- Partial 0716-2 RA – Central (riding) region reduced
- Partial 0912-1 RF – Central (riding) region reduced

Tuning Data; concatenation of:

- Partial 0813-1 LA (laboratory) – Subset from beginning including two mounts
- Partial 0912-2 RG – Central (riding) region reduced
- Partial 0913-3 RF – Central (riding) region reduced

Data from the riding phase was removed exactly as done for the initial model. **In addition the output from an earlier run of a smaller (less than 60 neurons) model was added to form a type of two tiered approach.**

Testing Data; concatenation of:

- Full 0719-1 RB
- Full 0812-2 RA
- Full 0829-1 RC
- Full 0830-3 RD
- Full 0902-1 RE
- Full 0902-2 RC

The entire data file was always used for testing files. The output from the same file, run through a smaller model was added as an additional column to the test files.

6.7.6 *Static Parameter Description (04-01)*

The following static parameters were used during this iteration:

- Error Cost Function: (conditional mean square error class + conditional mean square error non-class)$/2$
- Included Sensors: Gyroscope only
- The sensor readings are linearly transformed, real-values ranging between -1 and 1. No outliers were removed
- Additional column of data containing the output from a smaller ESN added to each file.

6.7.7 *PSO Parameter Description (04-01)*

PSO parameter ranges for search			
Name	Id	Value	Justification
Regularisation	α	0.0001 to 5	S. 6.5.7
Number of Neurons	N_x	100 to 360	S. 6.5.7
Input Scaling		0.0001 to 1	S. 6.5.7
Leaking Rate	a	0.0001 to 1	S. 6.5.7
Spectral Radius	$\rho(\mathbf{W})$	0.6000 to 1	S. 6.5.7
ESN Repeats		7	S. 6.5.7
Particles	S	14	S. 6.5.7
Maximum Generations		25	S. 6.5.7

Table 6.15: PSO Parameter Ranges for Iteration 04-01

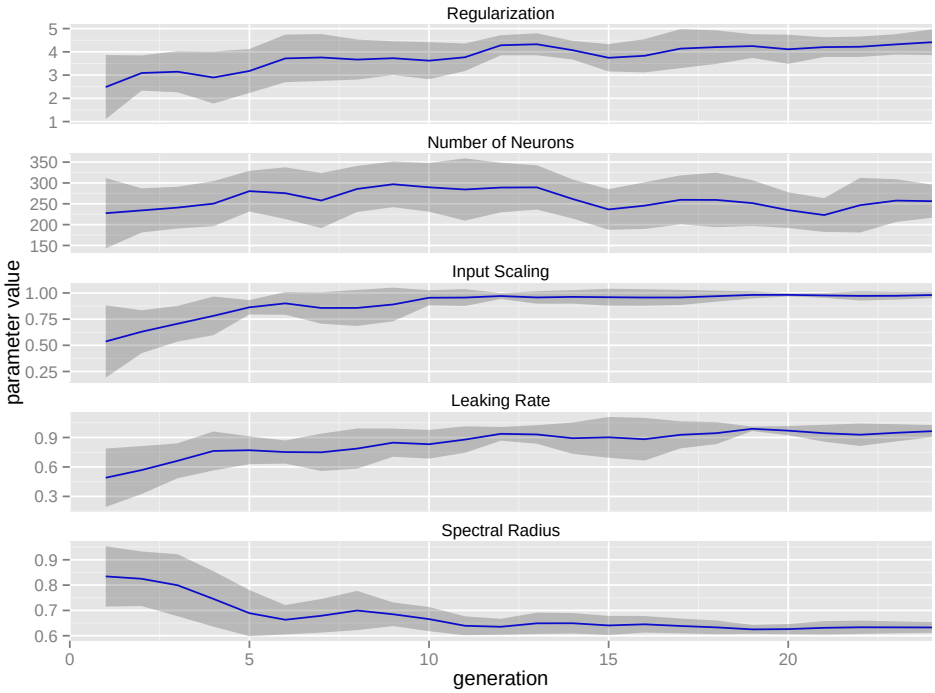


Figure 6.15: Parameter Optimisation for 04-01 Model

A PSO search was run using the parameter ranges listed in table 6.15. This search ran on an Amazon ECC virtual 8 processor machine running Debian. The ESN parameters with the lowest fitness score are listed in table 6.16. During parameter optimisation the Regularization parameter, unlike similar runs with different data

configurations, rose to a range between 4 and 5. Input Scaling took around 10 generations to drift towards its maximum value. Over around 12 generations, Leaking Rate stabilised at around 0.9 and Spectral Radius floated around 0.6. While the number of Neurons varied over quite a wide range centring around 250 throughout.

This was quite different to what happened with the parameters for the initial model. The only model parameter that remained similar was Input Scaling, Regression Regularisation changed from a minimal value to close to the maximum value permitted by the parameter bounds, Neuron Leak Rate changed from a low value to a high value while Spectral Radius and Number of Neurons both dropped by around a third.

6.7.8 *Run time Parameter Description (04-01)*

The parameters that produced the lowest error cost during optimisation are shown in table 6.16 and the ESN model that was built for this set of iterations was built, trained and tested on a Windows 7 based PC running an i7 processor and 8GB RAM with these parameters. This set of run-time parameters is markedly different from other sets within this series of iterations. Of particular note are:

- Number of neurons – This parameter normally gets optimised at somewhere near the limit of 360 but in this case settled at only 221.
- Neuron leak rate – In most other iterations this settles to around 0.1 to 0.2, which is a relatively slow leak rate. In this case it has settled close to the maximum and so this model will tend to have much less ”memory” than other models in the series.
- Regression Regularisation – Normally this settles at a very low level, close to the minimum and as a result minimal regularisation is done. In this case it is over 4.5 and close to its allowed maximum value, indicating that Regularisation was needed. This may well be as a result of a correlation between the values in the additive column and the test input.
- Spectral Radius – Normally this is close to its maximum value of 1.0 but in this case it has settled to 0.6.

These somewhat unexpected parameter values will likely provide a different ”view” from the ESN model. It is possible that the PSO search encountered a local minima that it failed to climb out of within the 25 generations that the PSO search was allowed to run. This parameter set was so unusual that the author searched back manually through the log that is kept from the PSO search and found an alternate parameter set that was closer to the expected values and this parameter set was saved for use in the next design iteration.

Table 6.16: Run Time Parameters for ESN iteration 04-01.

Description	**This Iteration**	Justification	Initial Iteration
Input values	4 **vectors of data**	Input Choice	3 vectors of data
Classes	**Mount only**	Class choice	Mount only
Number of neurons	**221**	Search	336
Input scaling	**0.9990**	Search	0.9252
Neuron leak rate	**0.9919**	Search	0.1379
Regularisation	**4.5198**	Search	0.0001
Spectral Radius	**0.6247**	Search	0.9455

6.7.9 *Training Results (04-01)*

The model was trained once using the best parameters found during the parameter optimisation process and with the training data specified, with resource usage and results as listed in table 6.17, with comparative figures for the initial model. The training error was more than twice as large (see table 6.17) as that from the initial model. Training completed in 3.7 seconds elapsed, this is less computational resources than the initial 02-01 model and is most likely as a result of the smaller number of neurons.

Table 6.17: Training Results for ESN iteration 04-01.

Description	**This Iteration**	Initial Iteration
User Time in seconds	**3.44**	7.28
System Time in seconds	**0.14**	0.09
Elapsed Time in seconds	**3.71**	7.50

6.7.10 *Testing Results (04-01)*

Table 6.18: Processing Times for Concatenated Datasets iteration 04-01.

Iteration Id	**04-01**	02-01
User Time in seconds	16.63	29.94
System Time in seconds	00.50	00.89
Elapsed Time in seconds	17.13	30.83

The area under the ROC curve has increased slightly for this ESN model, while the classification rate at the 0.6 cut-off level has fallen (Table 6.20). Looking at the ROC curve in Figure 6.19a, it can be seen that this ESN model is slightly more discriminative at very low model output levels (less than 0.02), slightly less discriminative between 0.02 and 0.1 and about the same from 0.1 and above.

Mount precision at the selected cut-off level has increased slightly, while Mount Recall has fallen with this additive model. There has been a noticeable drop in False Positives at 0.6 but that is juxtaposed against a noticeable drop in True Positives. There has also been an increase in the value of the ESN error cost function value for

Table 6.19: Classification Metrics for Concatenated Datasets iteration 04-01.

Iteration Id	**04-01**	02-01
ESN Test Error	0.2443	0.2046
RMSE	0.0620	0.0874
AUC	0.9617	0.9418

Table 6.20: Results for Concatenated Datasets iteration 04-01.

Measures at 0.6 Cut-Off				
Description	**04-01**		02-01	
Classification rate	0.571		0.714	
	Base Cl	**Mount Cl**	Base Cl	Mount Cl
Precision	0.9995	0.0830	0.9996	0.0737
Recall	0.9988	0.2124	0.9981	0.2920
Confusion Matrix at 0.6 Cut-Off				
	Predicted 04-01		Predicted 02-01	
Actual	**Base Cl**	**Mount Cl**	Base Cl	Mount Cl
Base Cl	217445	265	217295	415
Mount Cl	89	24	80	33

this model although not as pronounced as the increase seen during model training. Compute elapsed time has fallen substantially, most likely as a result of the smaller neuronal model.

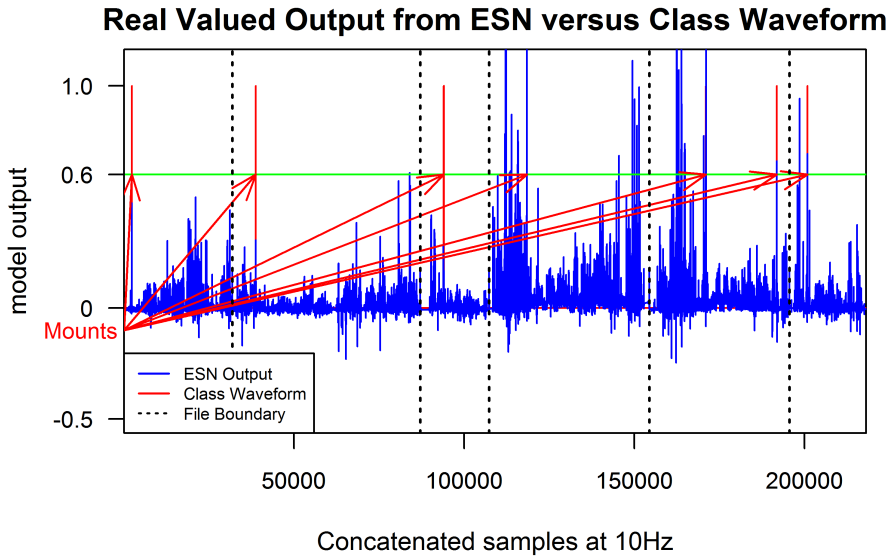
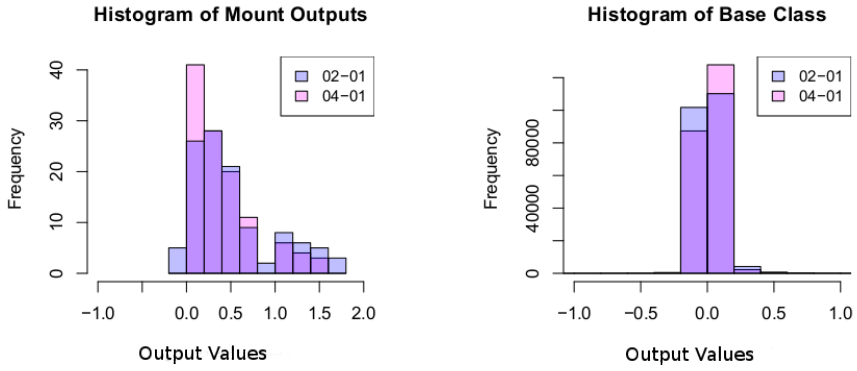


Figure 6.16: 04-01 - ESN Output for Concatenated Test Data

Looking at Figure 6.16 and comparing it with the equivalent figure for the initial 02-01 iteration, it looks as if the ESN output has mostly moved above zero for this ESN model but looking at the changes in the means (0.0210 compared with initial

0.0206) and standard deviations (0.0574 compared with initial 0.0852) it becomes apparent that while the mean has shifted slightly upwards, the main change is that there is considerably less variance with the current model. This reduced variance combined with a skewed frequency distribution anyway (see Figure 6.17) gives the impression that the response has tended to move above zero.

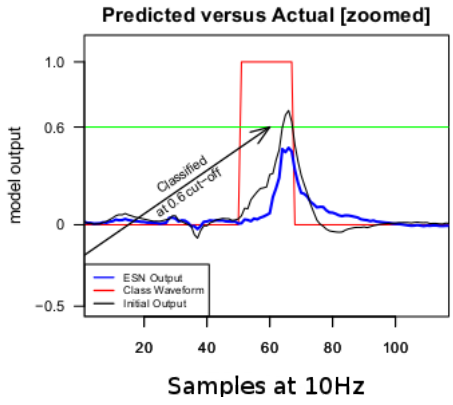


(a) Mount Class (b) Base Class

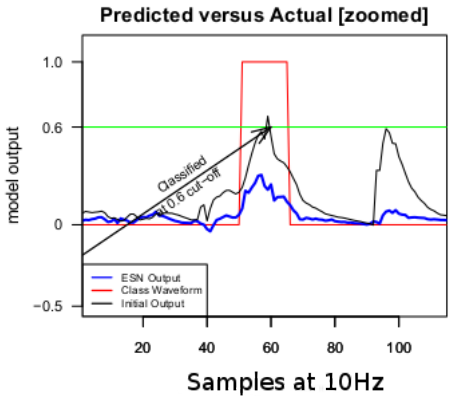
Figure 6.17: 04-01 - Comparing ESN Outputs 04-01 and 02-01

The Mounts shown in Figure 6.18 illustrate that the current model has managed to classify the 0902-2 RC participant session at the 0.6 cut-off level, which previous models have not done. However, in general, the ESN output from this model for Mount sequences has been more subdued with the mean Mount response falling from 0.5239 in the initial model to 0.4014 in this model and the standard deviation falling from 0.4561 to 0.3724. This more subdued response is probably responsible for the failure to classify both the 0719-1 RB and 0812-2 RA participant sessions.

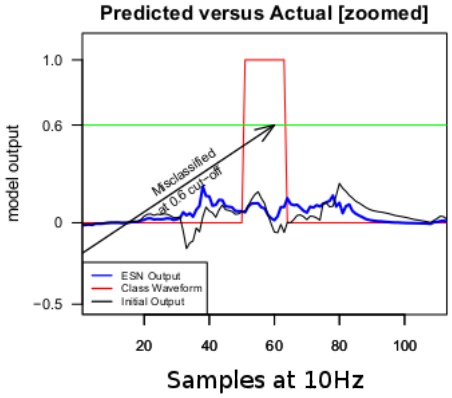
This model is unlike the earlier models. While the current model has slightly improved the area under the ROC curve, it has also produced worse results for the classification rate at the 0.6 cut-off. In addition, it failed to classify two Mount sequences that earlier model did classify but managed to classify another Mount sequence, 0902-2 RC, that earlier models failed to classify.



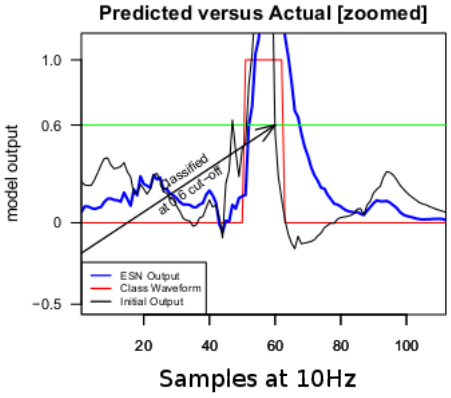
(a) 0719-1 RB #1



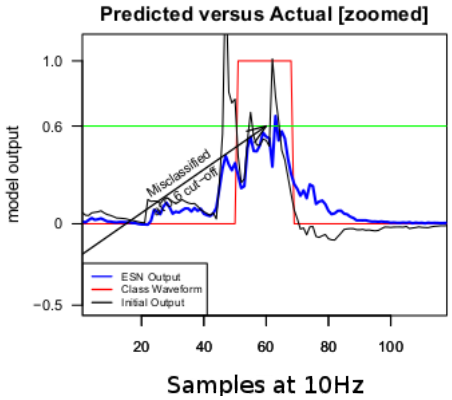
(b) 0812-2 RA #1



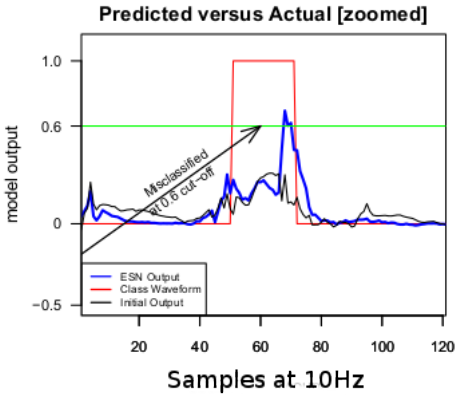
(c) 0829-2 RC #1



(d) 0830-3 RD #1

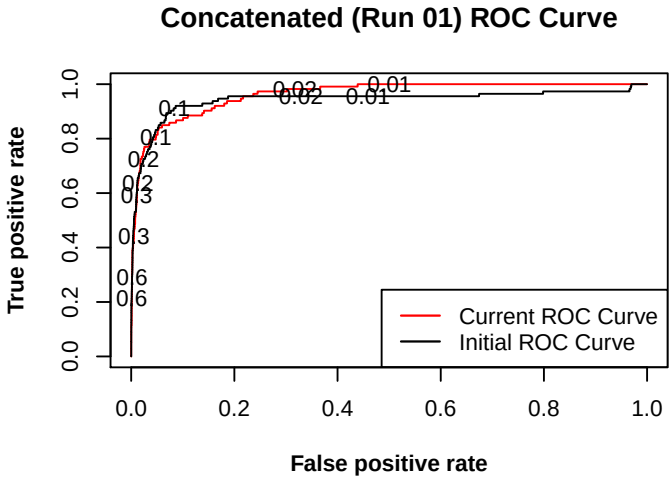


(e) 0902-1 RE #2

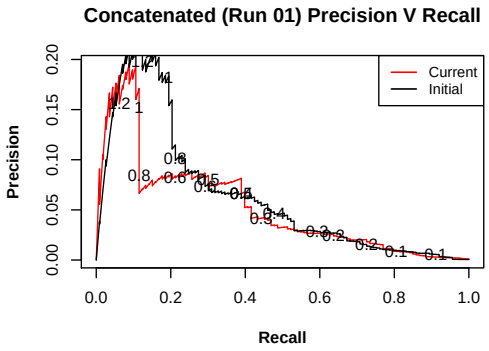


(f) 0902-2 RC #1

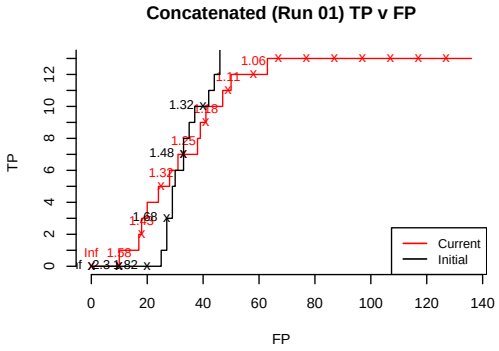
Figure 6.18: Mounts for Concatenated (04-01); 0902-1 RE #1 not shown



(a) ROC Curve



(b) Precision V Recall



(c) TP V FP

Figure 6.19: Performance Plots for Concatenated Test Data (04-01)

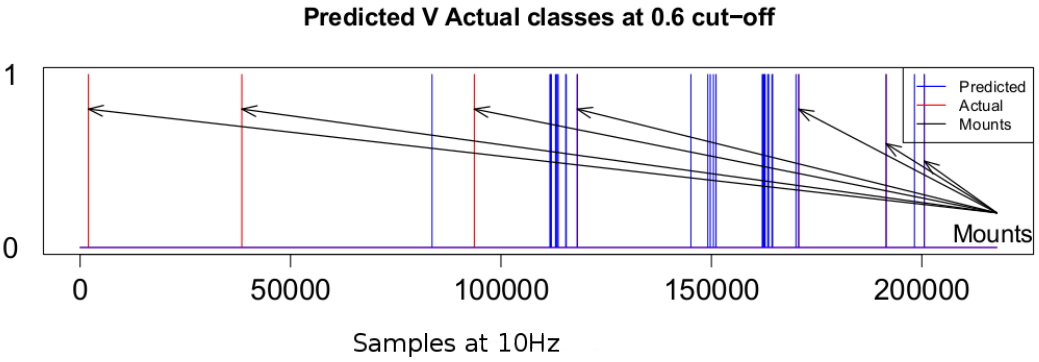


Figure 6.20: Classes for Concatenated Test Data (04-01)

6.7.11 *Discussion (04-01)*

Overall a mixed bag of results. The substantially different ESN model parameters consistently ran with less computing resources than the initial model, probably due to the number of neurons being 66% (221) less than those for the initial model (336).

The smaller number of neurons combined with the very high leak rate of 0.991919 will have resulted in there being much less memory effect from the recurrent reservoir connections and this has resulted in a different "view" on the input data.

Three participant sessions (0719-1 RB, 0812-2 RA and 0902-1 RE) produced results that were worse than those produced from the initial model. In particular, 0719-1 RB produced much worse results and failed to classify the Mount at a 0.6 cut-off.

At the same time, the remaining three participant sessions (0829-2 RC, 0830-3 RD and 0902-2 RC) produced results that were better than the initial model and in particular 0902-2 RC produced much better results and was able to classify the Mount at a 0.6 cut-off.

In all cases the number of False Positives were either substantially reduced (0830-3 RD, 0902-1 RE and 0902-2 RC) or remained the same at very low or zero numbers (0719-1 RB and 0812-2 RA and 0829-2 RC). This desirable characteristic is most likely a result of the current model producing output with less variance than the initial model (see table 6.21). The ESN output is skewed and any reduction in overall variance will tend to result in a reduction in the frequency of False Positives as the base class output will tend to be closer to zero.

Table 6.21: Comparison of Output Variance for iteration 04-01.

	Current		Initial	
Description	**Std Dev**	**Mean**	Std Dev	Mean
Concat Overall	**0.0574**	**0.0210**	0.0852	0.0206
Concat Mounts	**0.3724**	**0.4014**	0.4561	0.5239
Concat Base Class	**0.0561**	**0.0208**	0.0838	0.0204
0719-1 RB	**0.0330**	**0.0125**	0.0518	−0.0102
0812-2 RA	**0.0238**	**0.0039**	0.0409	0.0046
0829-2 RC	**0.0326**	**0.0044**	0.0452	−0.0019
0830-3 RD	**0.0858**	**0.0534**	0.1344	0.0709
0902-1 RE	**0.0653**	**0.0288**	0.0847	0.0264
0902-2 RC	**0.0368**	**0.0081**	0.0597	0.0062

6.7.12 *Results versus Goals (04-01)*

Based on the chosen measures, this model has produced a mixed result. It improves the area under the ROC curve but results in a lower classification rate at the 0.6 cut-off level. Looking at some of the individual participant sessions, the current model improves on the classification accuracy of the initial model in some cases and this characteristic plus the model's tendency to reduce or maintain a lower level of False

Positives is attractive as a possible alternative ”view” on the data to that of the initial model. It is not however, a substantial improvement over the initial model.

In addition, the ESN model parameters are so different from the optimised parameters obtained from other iterations in this series that it raises the possibility that the PSO search encountered a local minima in this case and was captured by that minima. With this possibility in mind, it was decided to re-run this iteration using the model parameters from a similar iteration to see if an additive model with more common parameters produces a better or different result. The results of this re-run are reported as iteration 04-02, following.

6.8 ENSEMBLE #1A - ESN MODEL WITH REAL WORLD DATA (04-02)

This iteration was similar to iteration 04–01 except that it built and tested the ESN model using parameters saved from the PSO output log from the prior iteration, as explained in Section 6.7.8. The saved parameter set was chosen rather than re-running the parameter optimisation process with a longer termination generation because the saved parameters were closer to the expected parameter values and also because running a longer optimisation process would take much longer and time to complete the research was quickly running out.

6.8.1 *Problem Identification (04-02)*

The last design iteration contained two sets of changes, the first to the run time parameters for the ESN model and the second to the input data that was processed by the model. Having two sets of changes meant that it was difficult to decide which set of changes, if only one, had resulted in the change to the classifiers ability to classify the activity of interest. A process is needed to isolate the input data changes so that the effects of changing the input data characteristics can be observed.

6.8.2 *Motivation (04-02)*

The motivation for this design iteration is to rerun the prior iteration but without run time parameters changes so that the change in input data characteristics can be observed.

6.8.3 *Design Goal (04-02)*

The goal of this design iteration is to repeat the work from 04-01 but using an relatively common parameter set to build and train the ESN model so that the effect of the input data change can be observed and to observe if the change in the input data has been beneficial for classification or not.

6.8.4 *Measure of Success (04-02)*

Either an overall improvement in the classification measures from 04-01 or an improvement over the classification measures from the initial model.

6.8.5 *Data Description (04-02)*

Training Data; concatenation of:

- Partial 0813-1 LA (laboratory) – Subset from end including two mounts
- Partial 0716-2 RA – Central (riding) region reduced
- Partial 0912-1 RF – Central (riding) region reduced

Tuning Data; concatenation of:

- Partial 0813-1 LA (laboratory) – Subset from beginning including two mounts
- Partial 0912-2 RG – Central (riding) region reduced
- Partial 0913-3 RF – Central (riding) region reduced

Data from the riding phase was removed exactly as done for the initial model. **In addition the output from an earlier run of a smaller (less than 60 neurons) model was added to form a type of two tiered approach.**

Testing Data; a concatenation of:

- Full 0719-1 RB
- Full 0812-2 RA
- Full 0829-1 RC
- Full 0830-3 RD
- Full 0902-1 RE
- Full 0902-2 RC

The entire data file was always used for testing files. The output from the same file, run through a smaller model was added as an additional column to the test files.

6.8.6 *Static Parameter Description (04-02)*

The following static parameters were used during this iteration:

- Error Cost Function: (conditional mean square error class + conditional mean square error non-class)$/2$
- Included Sensors: Gyroscope only
- The sensor readings are linearly transformed, real-values ranging between -1 and 1. No outliers were removed
- Additional column of data containing the output from a smaller ESN added to each file.

6.8.7 *Run time Parameter Description (04-02)*

The saved, alternate parameters from the prior iteration are shown in table 6.22 and the ESN model that was built for this set of design iterations was built, trained and tested on a Windows 7 based PC running an i7 processor and 8GB RAM with these parameters. The initial model (02-01) and 04-01 model parameters are also listed for comparative purposes.

Table 6.22: Run Time Parameters for ESN iteration 04-02.

Description	**04-02**	Justification	04-01	02-01
Input data	**4 vectors**	Input choice	4 vectors	3 vectors
Classes	**Mount only**	Class choice	Mount only	Mount only
Number of neurons	**294**	Alt. search	221	336
Input scaling	**0.9899**	Alt. search	0.9990	0.9252
Neuron leak rate	**0.2286**	Alt. search	0.9919	0.1379
Regularisation	**0.0004**	Alt. search	4.5198	0.0001
Spectral Radius	**0.8823**	Alt. search	0.6247	0.9455

6.8.8 *Training Results (04-02)*

The model was trained once using the parameters from table 6.22 and with the training data specified, with resource usage and results as listed in table 6.23. Training completed in 5.87 seconds elapsed. The training error at 0.1197 was comparable with that from the initial model and substantially less than the training error from 04-01.

Table 6.23: Training Results for ESN iteration 04-02.

Description	**04-02**	04-01	02-01
User Time in seconds	**5.58**	3.4400	7.28
System Time in seconds	**0.14**	0.1400	0.09
Elapsed Time in seconds	**5.87**	3.7100	7.50

6.8.9 *Test Results (04-02)*

Once training was done, each participants data set was concatenated into a single file and run against the trained, ESN model. Each individual data set was also used with the same model to try to find session specific nuances. The results of the concatenated file are reported along with summaries from the individual session tests.

Table 6.24: Processing Times for Concatenated Datasets iteration 04-02.

Iteration Id	**04-02**	04-01	02-01
User Time in seconds	25.54	16.63	29.94
System Time in seconds	00.65	00.50	00.89
Elapsed Time in seconds	26.20	17.13	30.83

Table 6.25: Classification Metrics for Concatenated Datasets iteration 04-02.

Iteration Id	**04-02**	04-01	02-01
ESN Test Error	0.2745	0.2443	0.2046
RMSE	0.0771	0.0620	0.0874
AUC	0.8714	0.9617	0.9418

Table 6.26: Results for Concatenated Datasets iteration 04-02.

Measures at 0.6 Cut-Off						
Description	**04-02**		04-01		02-01	
Classification Rate	0.714		0.571		0.714	
	Base	**Mount**	Base	Mount	Base	Mount
Precision	0.9996	0.1039	0.9995	0.0830	0.9996	0.0737
Recall	0.9982	0.2743	0.9988	0.2124	0.9981	0.2920
Confusion Matrix at 0.6 Cut-Off						
	Predicted 04-02		Predicted 04-01		Predicted 02-01	
Actual	**Base**	**Mount**	Base	Mount	Base	Mount
Base	217503	207	217445	265	217295	415
Mount	89	24	89	24	80	33

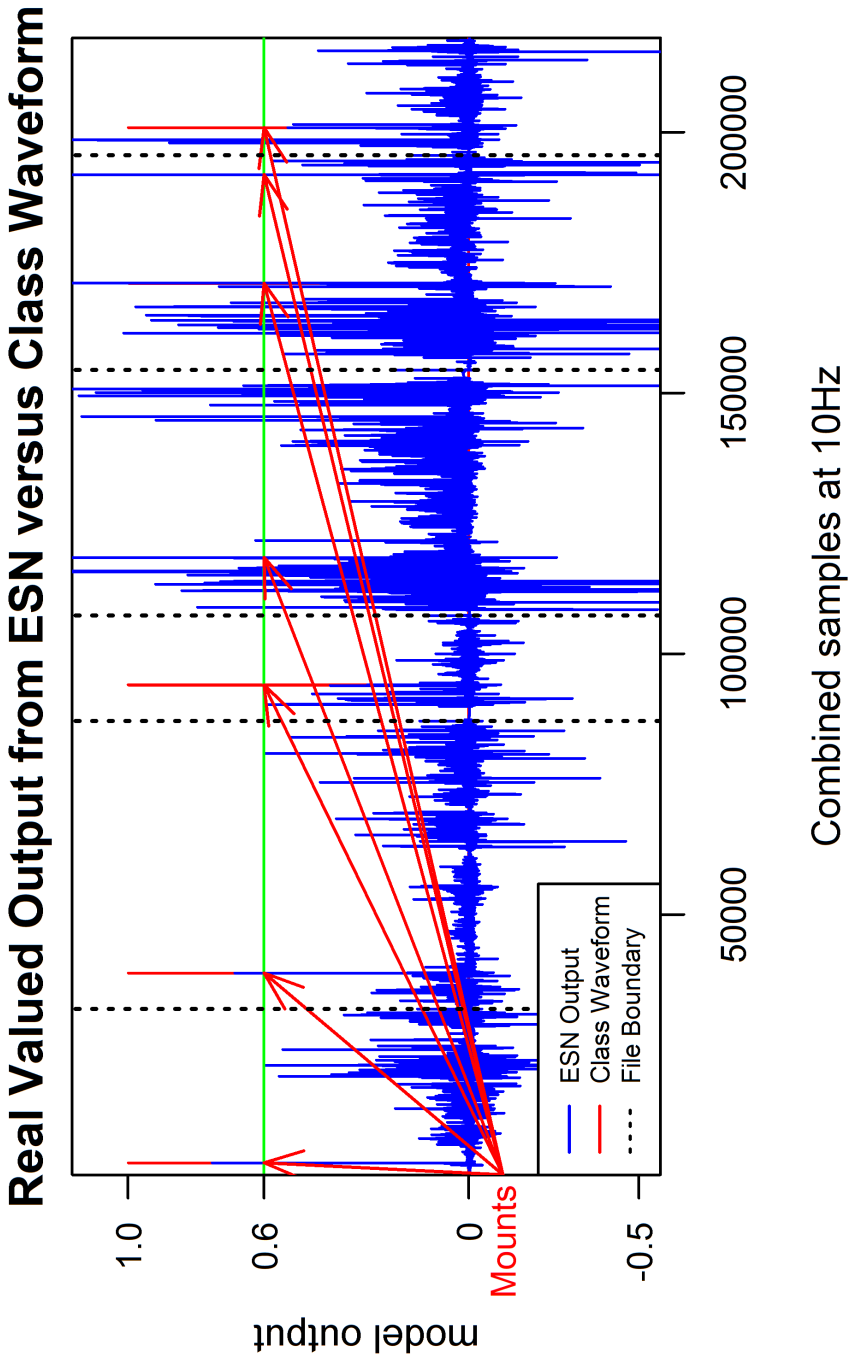
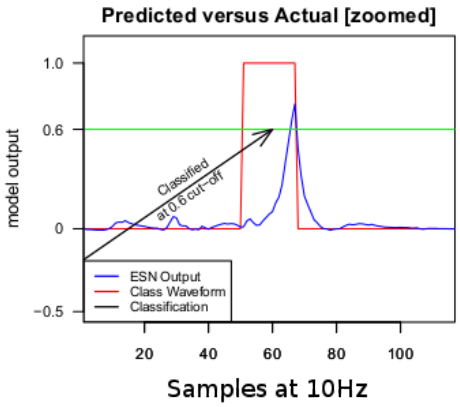
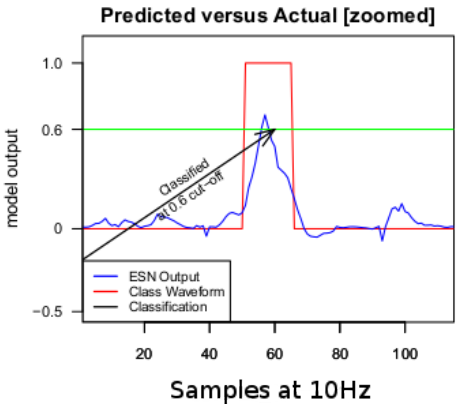


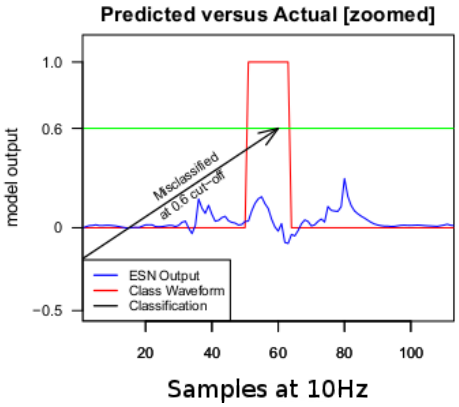
Figure 6.21: Concatenated ESN Output for 04-02



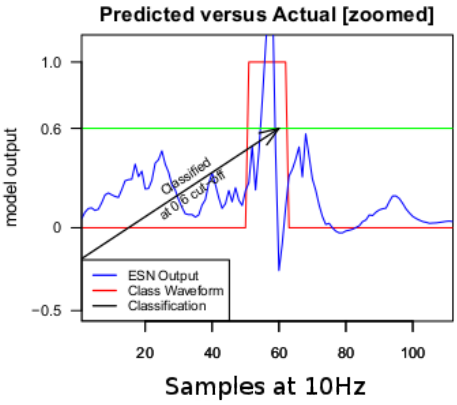
(a) 0719-1 RB #1



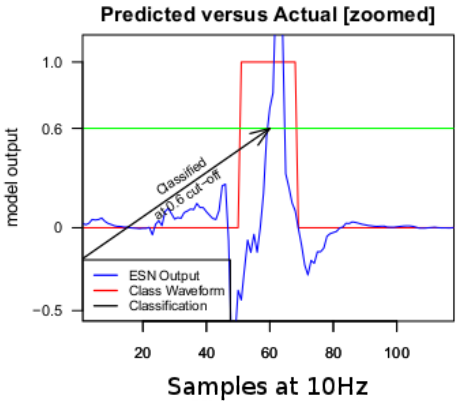
(b) 0812-2 RA #1



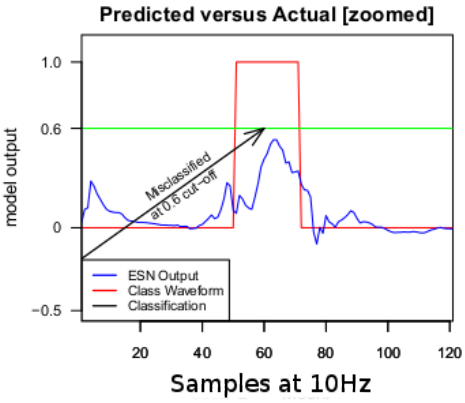
(c) 0829-2 RC #1



(d) 0830-3 RD #1

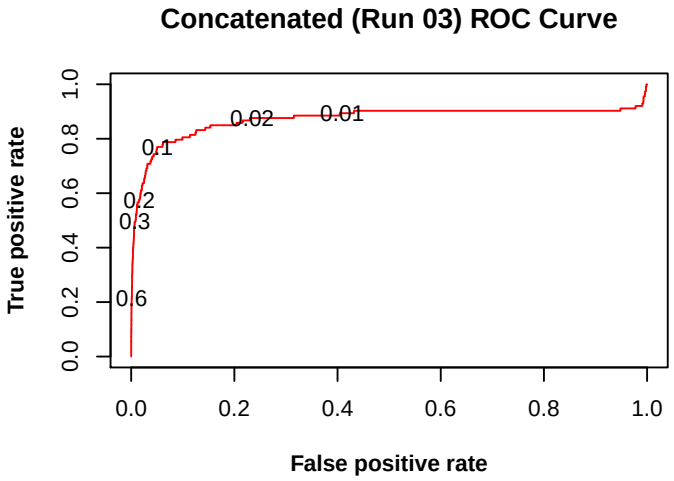


(e) 0902-1 RE #2

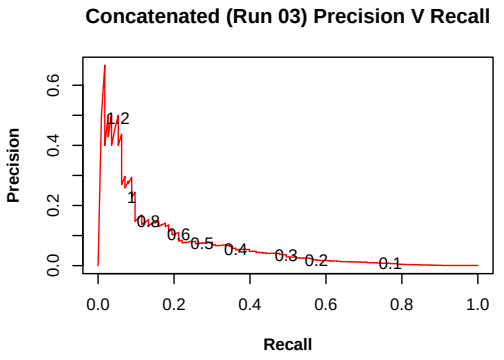


(f) 0902-2 RC #1

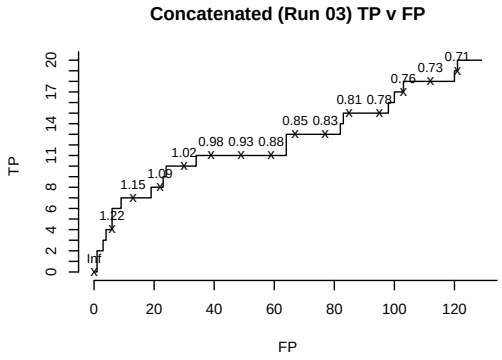
Figure 6.22: Concatenated Mounts (04-02); 0902-1 RE #1 not shown



(a) ROC Curve



(b) Precision V Recall



(c) TP V FP

Figure 6.23: Concatenated Datasets Performance Plots (04-02)

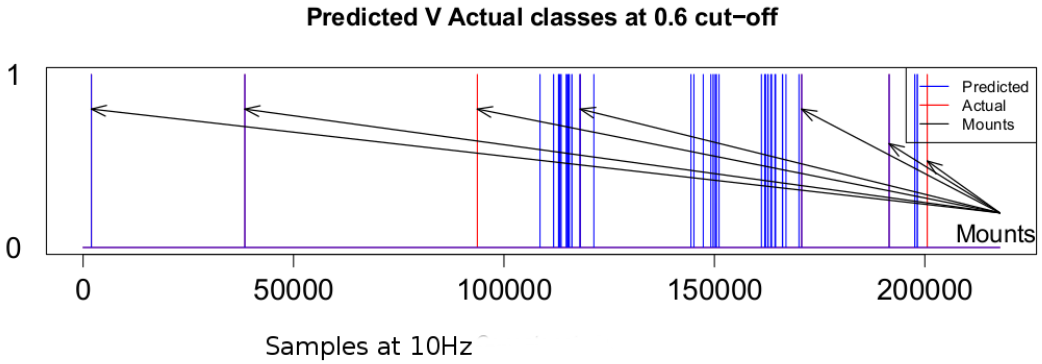


Figure 6.24: Concatenated datasets Classes (04-02)

6.8.10 *Discussion (04-02)*

This set of tests for the 04-02 ESN model once again produced mixed results and it became clearer that simple one on one comparisons between results from each data set was not sufficient to provide a reliable comparison of one model against another.

Table 6.27: Comparing Precision Across Four Design Iterations.

Precision at 0.6 Cut-Off				
Data-set	02-01	03-01	04-01	04-02
0719-1 RB	1.0000	1.0000	0.0000	1.0000
0812-2 RA	0.5000	1.0000	0.0000	1.0000
0829-2 RC	0.0000	0.0000	0.0000	0.0000
0830-3 RD	0.0263	0.0194	0.0529	0.0301
0902-1 RE	0.1875	0.1456	0.1325	0.2586
0902-2 RC	0.0000	0.0000	0.1875	0.0000
Mean	0.2856	0.3608	0.0622	0.3815
SD	0.3991	0.4981	0.0804	0.4888

Table 6.28: Comparing Recall Across Four Design Iterations.

Recall at 0.6 Cut-Off				
Data-set	02-01	03-01	04-01	04-02
0719-1 RB	1.0000	1.0000	0.0000	1.0000
0812-2 RA	0.5000	1.0000	0.0000	1.0000
0829-2 RC	0.0000	0.0000	0.0000	0.0000
0830-3 RD	0.0263	0.0194	0.0529	0.0301
0902-1 RE	0.1875	0.1456	0.1325	0.2586
0902-2 RC	0.0000	0.0000	0.1875	0.0000
Mean	0.2856	0.3608	0.0622	0.3815
SD	0.3990	0.4981	0.0804	0.4888

The substantially different ESN model parameters consistently ran with less computing resources than the initial model, probably due to the number of neurons being 66% (221) less than those for the initial model (336).

The smaller number of neurons combined with the very high leak rate of 0.991919 will have resulted in there being much less memory effect from the recurrent reservoir connections and this has resulted in a different ”view” on the input data.

Three data sets run against this model (0719-1 RB, 0812-2 RA and 0902-1 RE) produced results that were worse than those produced from the initial model. In particular, 0719-1 RB produced much worse results and failed to classify the Mount at a 0.6 cut-off.

At the same time, the remaining three data sets (0829-2 RC, 0830-3 RD and 0902-2 RC) produced results that were better than the initial model and in particular 0902-2 RC produced much better results and was able to classify the Mount at a 0.6 cut-off.

In all cases the number of False Positives were either substantially reduced (0830-3 RD, 0902-1 RE and 0902-2 RC) or remained the same at very low or zero numbers (0719-1 RB and 0812-2 RA and 0829-2 RC). This desirable characteristic is most likely a result of the current model producing output with less variance than the initial model (see table 6.29). The classes are so skewed towards the base class that any reduction in overall variance will result primarily from a reduction in variance of the base class and as the base class is represented by a zero output, such a reduction will tend to reduce the frequency of False Positives as the base class output will tend to be closer to zero.

Table 6.29: Comparison of Output Variance for iteration 04-02.

	Current		Initial	
Data set name	**Std Dev**	**Mean**	Std Dev	Mean
0719-1 RB	0.0330	0.0125	0.0518	−0.0102
0812-2 RA	0.0238	0.0039	0.0409	0.0046
0829-2 RC	0.0326	0.0045	0.0452	−0.0020
0830-3 RD	0.0858	0.0534	0.1344	0.0710
0902-1 RE	0.0653	0.0288	0.0848	0.0264
0902-2 RC	0.0368	0.0081	0.0597	0.0062

6.8.11 *Results versus Goals (04-02)*

It is difficult to state empirically that the current additive ESN model is better or worse than the initial model because of the mixed results. However, the current model improves on the classification accuracy of the initial model in some cases and this characteristic plus the model's tendency to reduce or maintain a lower level of False Positives is attractive as a possible alternative "view" on the data to that of the initial model.

6.8.12 *Conclusions (04-02)*

The classification ability of the ESN model was much more in line with iteration 02–01 and 03–01 when set up and tested with this alternate parameter set and it was not otherwise, particularly remarkable. This design iteration, together with the prior one (04-01), caused a re-think of the use of a parameter optimisation process to search the parameter space in conjunction with changes to the input data. In essence, an attempt was being made to compare a number of iterations of similar models whilst changing two sets of important variables (the data characteristics and the model parameter characteristics). While such an approach might, eventually, lead to an optimum position the process of getting there would be complex. A decision was made to change tack slightly and to only change one set of variables at a time. A decision was made to initially keep the model parameter variables static while exploring data changes and then if and when a recommended data configuration was standardised then to go back to exploring the model parameters. A decision was made to standardise on the

model parameters from iteration 02-01 to prevent the necessity of having to go back and redo prior work. The parameters from 02-01 had been found to be reasonably comparable with other experiments with the exception of 04-01 and 02-01 was the benchmark against which newer design iterations were compared.

A decision was also made to change the way future design iterations were reported. With the removal of the model parameter optimisation process, reporting the results of the optimisation become pointless. This reduces the amount of detail that needs to be reported. In the spirit of reporting simplification, a decision was also made to stop reporting classification results for individual riding sessions and instead to only report the results of classifying the entire, concatenated test dataset. Together, these decisions will result in much more compact reporting of future work.

In addition, a decision was made to only report the results of design iterations where substantial progress was made. Where an individual iteration did not result in substantial change, it would only get mentioned in passing. In conjunction with this, an additional cross-reference identifier will be reported that allows the results of a design iteration to be cross-referenced with the exact code that was run to implement that design iteration.

6.9 UNDER SAMPLING - ESN WITH REAL WORLD DATA (05-01 V13)

Under-sampling is an accepted technique for improving classification rates when dealing with rare classes (Chawla, 2005; Drummond, Holte, & others, 2003; Weiss, 2004, 2010). Each design iteration within this research since (and including) iteration 02-01 has used under-sampling to reduce the data within the central, riding region of the recorded activities. In this series of iterations the amount of under-sampling is progressively increased by including less of the central, riding region from each dataset. The first design iteration in this series (V11) under-sampled by including all data from the start of each dataset up until 4,000 past the mount and then concatenated this with data from 4,000 samples prior to the dismount through to the end of the dataset. Essentially this reduced the number of samples used during riding time while keeping all other samples. Of course, this under-sampling was only done to the training data, no data was excluded from the test datasets. The second design iteration (V12) increased the under-sampling by including only up to 1,000 samples after mounting and 1,000 samples prior to dismounting. The third iteration (V13) kept the under-sampling rate the same as V12 but took advantage of no longer needing to tune the ESN model parameters and so was able to include the data previously used during tuning to add to the amount of training data available. The rationale for excluding large parts of the data captured after mounting and prior to dismounting was that during prior real-world design iterations it had been noted that most false positives were occurring either prior to mounting or after dismounting but not in between these events.

As noted in section 6.8.12, this set of design iterations follows a slightly different approach by using a static set of ESN model parameters copied from iteration 02-01 and aims to find the highest classification result from a small set of changes to how the input data is under–sampled prior to classification. Only three different design iterations were run and tested, identified as V11 through V13, and the best performing iteration, V13, is described in detail with any important factors from the other two noted in passing. Please note that this is not a comprehensive set of tests to provide insight into the best possible way to under-sample the input data prior to classification as the large scale of such a goal is outside of this research but is instead a way to try a limited set of ideas relating to under–sampling.

6.9.1 *Problem Identification (05-01)*

Earlier instantiations of the ESN classification engine have demonstrated a good ability to classify *mounts* based on real-world data but while these initial results are quite impressive they are not “good enough” to use in real life. Earlier instantiations have thrown out too many False Positives and have not successfully classified all of the mounts. Essentially the discriminative ability of the ESN classification engine needs to be improved.

6.9.2 *Motivation (05-01)*

The motivation for this design iteration was to explore some aspects of under-sampling as an aid to improving the ability of the ESN classification engine to discriminate the activity of interest.

6.9.3 *Design Goal (05-01)*

The goal of this set of design iterations was to improve the classification ability of the ESN model by altering the characteristics of the input data.

6.9.4 *Measure of Success (05-01 - V13)*

An overall improvement in the area under the ROC curve while maintaining or improving the classification rate as defined in section 6.5.10.

6.9.5 *Data Description (05-01 - V13)*

Training Data; concatenation of:

- Partial 0813-1 LA (laboratory) – Subset from end including two mounts
- Partial 0716-2 RA – Central (riding) region reduced
- Partial 0912-1 RF – Central (riding) region reduced
- Partial 0912-2 RG – Central (riding) region reduced
- Partial 0913-3 RF – Central (riding) region reduced

Data from the riding phase was removed from each dataset in larger amounts compared with the initial 02-01 model. Only two sections of 1,000 samples of riding phase data were included from each dataset, 1,000 from immediately after the mount and 1,000 immediately preceding the dismount.

Testing Data; concatenation of:

- Full 0719-1 RB
- Full 0812-2 RA
- Full 0829-1 RC
- Full 0830-3 RD
- Full 0902-1 RE
- Full 0902-2 RC

The entire data file was always used for testing files.

6.9.6 *Static Parameter Description (05-01 - V13)*

The following static parameters were used during this iteration:

- Error Cost Function: (conditional mean square error class + conditional mean square error non-class)$/2$
- Included Sensors: Gyroscope
- The Gyroscope sensor readings are linearly transformed, real-values ranging between 0 and 1.
- The Gyroscope data was diff'ed to ensure that the data mean was stationary
- No outliers were removed

6.9.7 *Run time Parameter Description (05-01 - V13)*

The parameters from iteration 02-01 were used for this design iteration and are shown in table 6.30 and the ESN model that was built for this set of design iterations was built, trained and tested on a Windows 7 based PC running an i7 processor and 8GB RAM.

Table 6.30: Run Time Parameters for ESN iteration 05-01.

Description	**This Iteration**	Justification	02-01 Iteration
Input values	3 vectors of data	Input choice	3 vectors of data
Classes	Mount only	Class choice	Mount only
Number of neurons	336	From 02-01	336
Input scaling	0.9252	From 02-01	0.9252
Neuron leak rate	0.1379	From 02-01	0.1379
Regularisation	0.0001	From 02-01	0.0001
Spectral Radius	0.9455	From 02-01	0.9455

6.9.8 *Training Results (05-01 - V13)*

The training error for this iteration was more than three times as large (see table 6.31) as that from the initial model. Training completed in 14.39 seconds elapsed, this is much more computational resources than the initial 02-01 model. The additional computational requirements are as a result of using the larger training file.

Table 6.31: Training Results for ESN iteration 05-01.

Description	**This Iteration**	02-01 Iteration
User Time in seconds	**13.98**	7.28
System Time in seconds	**0.24**	0.09
Elapsed Time in seconds	**14.39**	7.50

6.9.9 *Testing Results (05-01 - V13)*

Table 6.32: Processing Times for Concatenated Test Datasets iteration 05-01 V13.

Iteration Id	**05-01**	02-01
User Time in seconds	31.43	29.94
System Time in seconds	00.97	00.89
Elapsed Time in seconds	32.41	30.83

Table 6.33: Classification Metrics for Concatenated Test Datasets iteration 05-01 V13.

Iteration Id	**05-01**	02-01
ESN Test Error	0.3266	0.2046
RMSE	0.0529	0.0874
AUC	0.9062	0.9418

Table 6.34: Results for Concatenated Test Datasets iteration 05-01 V13.

Measures at 0.6 Cut-Off				
Description	**05-01**		02-01	
Classification rate	0.286		0.714	
	Base Cl	**Mount Cl**	Base Cl	Mount Cl
Precision	0.9995	0.1765	0.9996	0.0737
Recall	0.9998	0.0797	0.9981	0.2920
Confusion Matrix at 0.6 Cut-Off				
	Predicted 05-01		Predicted 02-01	
Actual	**Base Cl**	**Mount Cl**	Base Cl	Mount Cl
Base Cl	217668	104	217295	415
Mount Cl	42	9	80	33

The area under the ROCcurve increased (0.8676,0.8721 and 0.9062) across all three under-sampling iterations with this iteration (V13) having the largest area, however, this is still substantially less than the area under the ROC curve for the original, 02-01 iteration. In addition, the classification rate at the 0.6 cut-off level has fallen (Table 6.34) substantially. Looking at the ROC curve in Figure 6.27a, this ESN model is slightly more discriminative at low model output levels (less than 0.1) and about the same from 0.1 and above.

Mount precision at the selected cut-off level has increased, while Mount Recall has fallen with this model. There has been a substantial drop in False Positives at the 0.6 cut–off (104 versus 415) but that is juxtaposed against a large drop in both True Positives (9 versus 33) and in the Classification Rate (0.286 versus 0.714).

Looking at Figure 6.25 and comparing it with the equivalent figure for the initial 02-01 iteration (Figure 6.5 on page 155), the ESN output for this iteration is considerably more subdued than 02-01 with much less variation and that this accounts

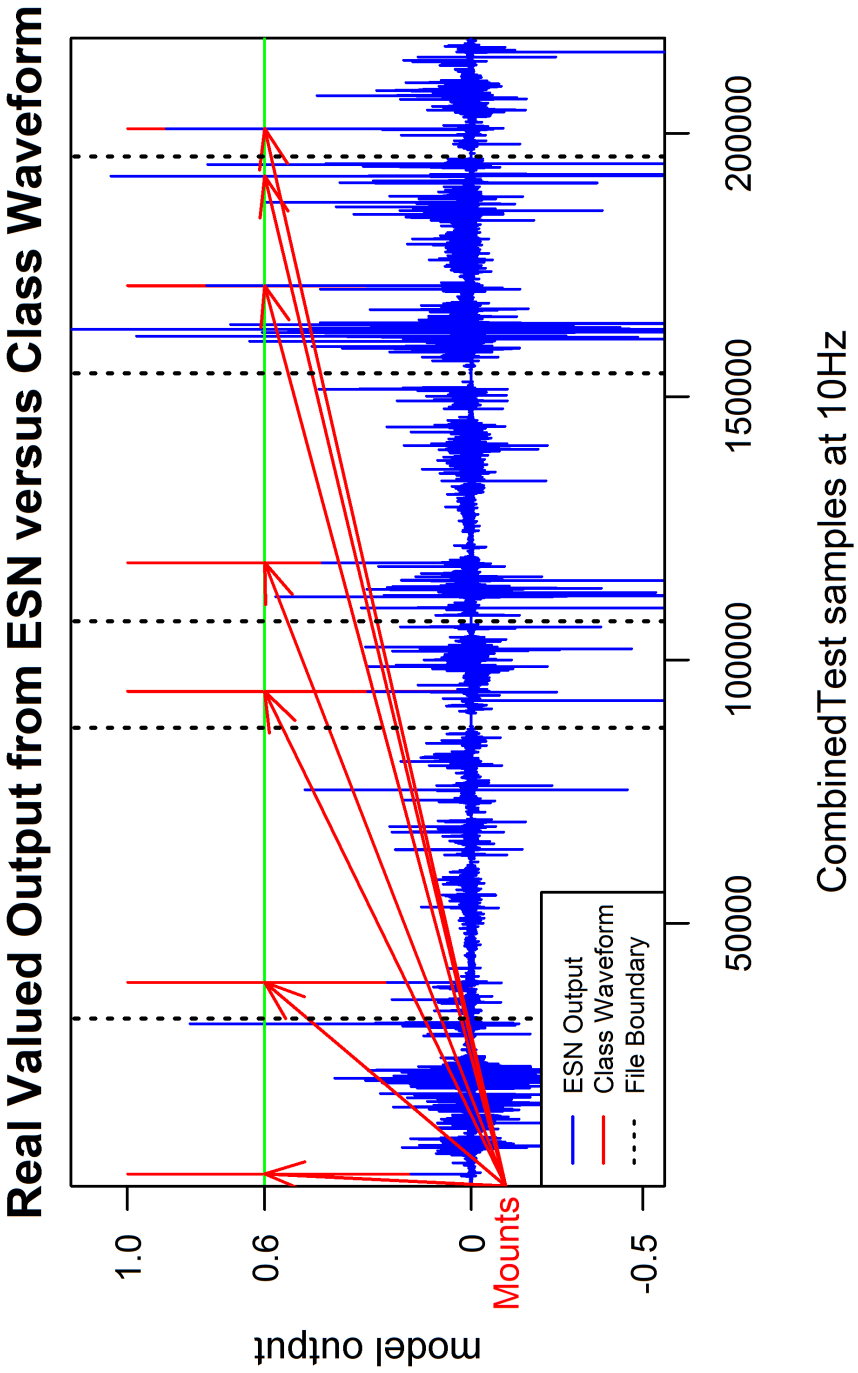
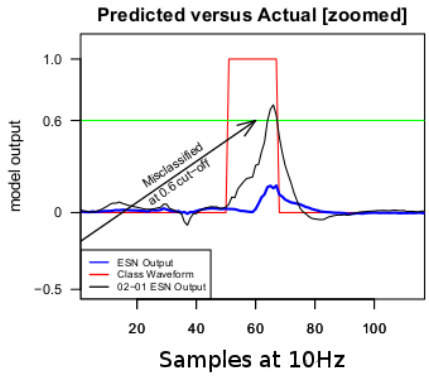
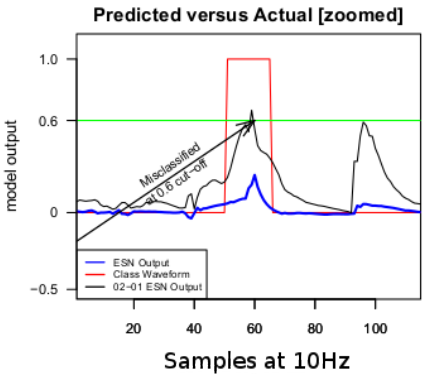


Figure 6.25: 05-01 V13 ESN Output for Concatenated Test Data

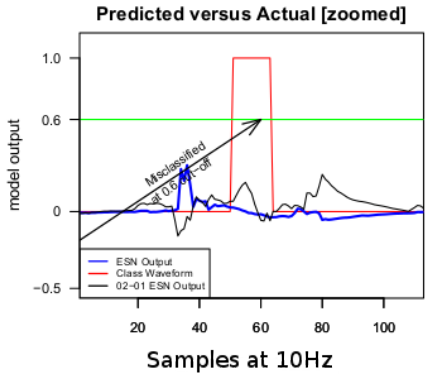
for both the drop in false positives and the drop in true positives. The ESN model response for iteration 05-01 V13 has tended to cluster around zero.



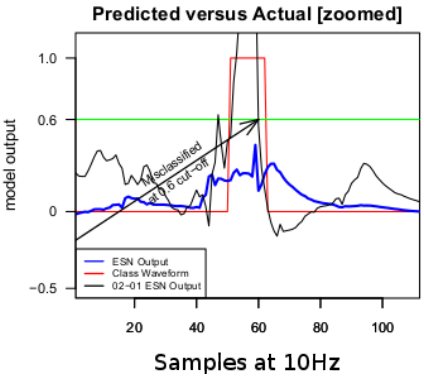
(a) 0719-1 RB #1



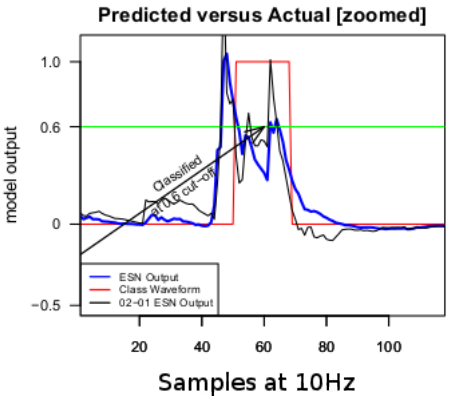
(b) 0812-2 RA #1



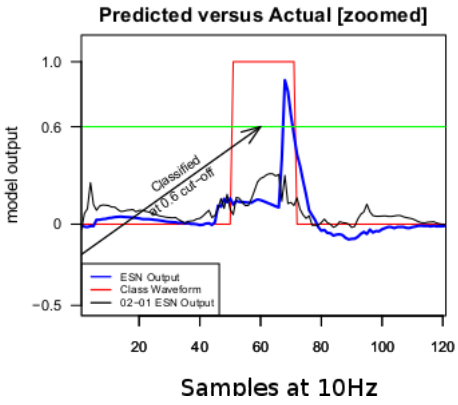
(c) 0829-2 RC #1



(d) 0830-3 RD #1

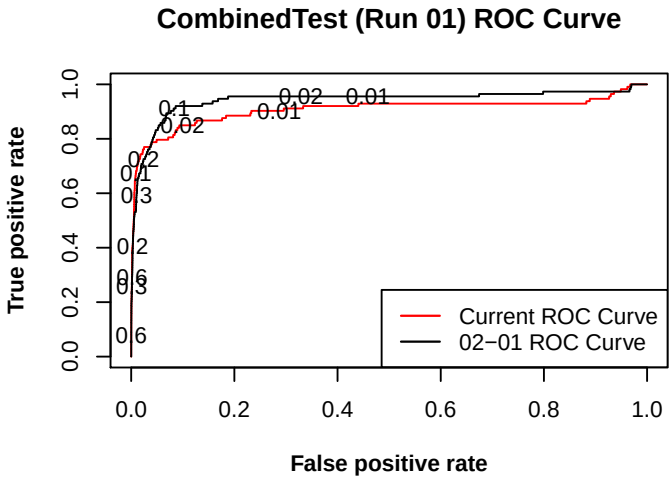


(e) 0902-1 RE #2

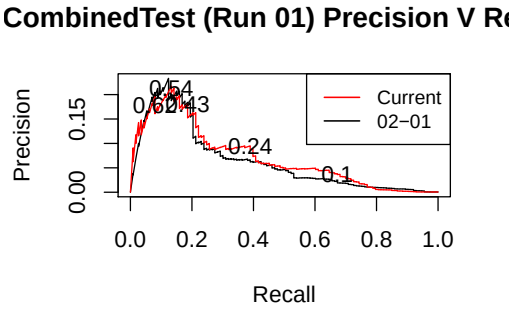


(f) 0902-2 RC #1

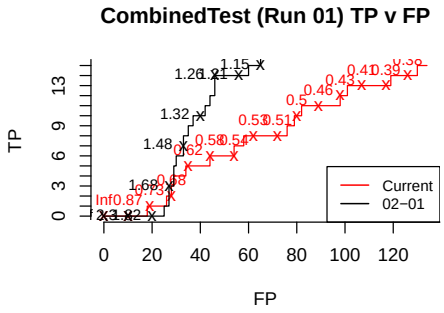
Figure 6.26: 05-01 V13 Mounts for Concatenated; 0902-1RE #1 not shown



(a) ROC Curve



(b) Precision V Recall



(c) TP V FP

Figure 6.27: 05-01 V13 Performance Plots for Concatenated Test Data

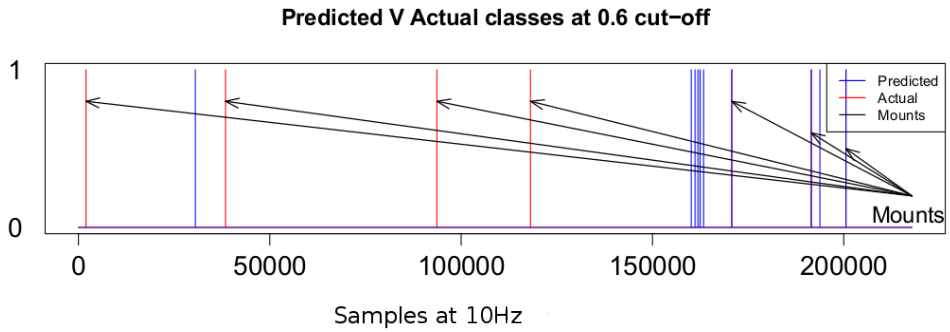


Figure 6.28: 05-01 V13 Classes for Concatenated Test Data

6.9.10 *Results versus Goals (05-01) V13*

This iteration did not meet either of the design goals. The area under the ROC curve for this design iteration is less than the goal and the classification rate is also less than the goal.

6.9.11 *Discussion (05-01) V13*

At first look this design iteration has produced worst results than the initial ESN model with a substantially lower classification rate, however, this model produces substantially less False Positives and so if it is possible to retain this ability to reduce False Positives while somehow boosting the model's response to True Positives then under-sampling may yet prove useful.

6.10 BUTTERWORTH - ESN WITH REAL WORLD DATA 06-01 V15

This set of design iterations uses a static set of ESN model parameters copied from iteration 02-01, as noted in section 6.8.12, and aims to find the highest classification result from a small set of changes to how the input data is filtered prior to classification. Only three different design iterations were run and tested, identified as V14 through V16, and the best performing iteration, V15, is described in detail with any important factors from the other two noted in passing. Please note that this is not a comprehensive set of tests to provide insight into the best possible way to filter the input data prior to classification as the large scale of such a goal is outside of this research but is instead a way to try a limited set of ideas relating to filtering.

Filtering is an accepted technique for improving classification rates by removing very low frequency trends or drift and high frequency noise from within the accelerometer and gyroscope data and is recommended by a number of authors including Anguita, Ghio, Oneto, Parra, and Reyes-Ortiz (2013); Avci et al. (2010); K. Y. Chen and Bassett (2005); Davey, Anderson, and James (2008); Godfrey, Bourke, Olaighin, van de Ven, and Nelson (2011); Godfrey, Conway, Meagher, and ÓLaighin (2008); Lau and Tong (2008); Mannini and Sabatini (2010); Mathie et al. (2004); Plötz (2010). The question then arose of where the high pass cut off frequency should be set for filtering out the gyroscope drift. Mathie et al. (2004) reported that the lower bounds for most human activities sits at around 0.3Hz. C. Yang and Hsu (2010), while describing a particular inertial recording device suggested that a 0.25Hz lower bound was acceptable. Godfrey et al. (2008) suggested a lower bound of 0.6Hz but their work concentrated on human gait analysis and so is only partly comparable. K. Y. Chen and Bassett (2005) was the most specific when they recommended the use of a high pass filter set at 0.1Hz to eliminate gyroscope drift and so this was the value that we used for this set of design iterations. The next consideration was the type of filter to use and its configuration. D. Liu et al. (2014); Mannini and Sabatini (2010) both recommended a second order Butterworth filter in similar situations and so this is the filter configuration used. Lastly, a way of filtering the high frequency noise was considered. We could have modified the high pass Butterworth filter to become a band pass pass filter to deal with both the low frequency drift and the higher frequency noise but the author had not used a band pass filter before while a rolling mean filter had been used successfully during the exploratory phase of this work and so it was decided to use a rolling mean rather than either another, low pass filter or a band pass filter. In making this decision the author was cognizant of the design strategies "*favour simplicity*" and "*be mindful of possible implementation issues*". In the author's opinion it is (and was) simpler to filter the high frequency elements using a simple rolling mean and such an approach is computationally simpler and less demanding if and when implemented into wearable hardware.

It is worth noting here that prior design iterations within this research, except for the early LSM and ESN iterations within Chapter 5 (see page 115) where outliers were removed, have not used filtering techniques on the input data.

In this series of iterations the first iteration (V14) uses a relatively standard Butterworth filter only and obtained reasonable but not outstanding results with the area

under the ROC curve of 0.88716. The second iteration (V15) used both a Butterworth filter and a rolling mean and achieved very good results (area under the curve of 0.9694, better than the results from any prior iteration. The last iteration (V16) used the same filter and rolling mean as V15 but varied the ESN model parameters to try the parameters from iteration 04-01 where the unusual parameters resulted in an alternate view of the input data. Iteration V16 resulted in substantially worse results from classification (AUC of 0.84581) and so iteration V15 was adopted as the best iteration and used in subsequent design iterations. Design iteration V15 is described in the following sections in more detail.

6.10.1 *Problem Identification (06-01)*

Earlier instantiations of the ESN classification engine have demonstrated a good ability to classify *mounts* based on real-world data but while these initial results are quite impressive they are not "good enough" to use in real life. Earlier instantiations have thrown out too many False Positives and have not successfully classified all of the mounts. Essentially the discriminative ability of the ESN classification engine needs to be improved.

6.10.2 *Motivation (06-01)*

The motivation for this design iteration was to explore some aspects of filtering as an aid to improving the ability of the ESN classification engine to discriminate the activity of interest.

6.10.3 *Design Goals (06-01 - V15)*

The goals for this design iteration are:

1. Maintain or improve the classification rate from 0.714, as achieved with iteration 02-01.
2. Improve the area under the ROC curve from 0.94184, as achieved with iteration 02-01.
3. Maintain or reduce the number of false positives from 104, as achieved with iteration 05-01 V13.

6.10.4 *Measure of Success (06-01 - V15)*

An overall improvement in the area under the ROC curve while maintaining or improving the classification rate as defined in section 6.5.10 and a reduction in the number of False Positives.

6.10.5 *Data Description (06-01 - V15)*

Training Data; concatenation of:

- Partial 0813-1 LA (laboratory) – Subset from end including two mounts
- Partial 0716-2 RA – Central (riding) region reduced
- Partial 0912-1 RF – Central (riding) region reduced
- Partial 0912-2 RG – Central (riding) region reduced
- Partial 0913-3 RF – Central (riding) region reduced

Data from the riding phase was removed from each dataset in larger amounts compared with the initial 02-01 model. Only two sections of 1,000 samples of riding phase data were included from each dataset, 1,000 from immediately after the mount and 1,000 immediately preceding the dismount.

Testing Data; concatenation of:

- Full 0719-1 RB
- Full 0812-2 RA
- Full 0829-1 RC
- Full 0830-3 RD
- Full 0902-1 RE
- Full 0902-2 RC

The entire data file was always used for testing files.

6.10.6 *Static Parameter Description (06-01 - V15)*

The following static parameters were used during this iteration:

- Error Cost Function: (conditional mean square error class + conditional mean square error non-class)$/2$
- Included Sensors: Gyroscope
- The Gyroscope sensor readings are linearly transformed, real-values ranging between 0 and 1.
- The Gyroscope data was passed through a second order high-pass Butterworth filter with a frequency of 0.01Hz, designed to remove signal drift.
- The detrended Gyroscope data was then transformed with a rolling mean.
- No outliers were removed, however the rolling mean would tend to dampen the outliers.

6.10.7 *Run time Parameter Description (06-01 V15)*

The parameters from iteration 02-01 were used for this design iteration and are shown in table 6.35 and the ESN model that was built for this set of design iterations was built, trained and tested on a Windows 7 based PC running an i7 processor and 8GB RAM.

Table 6.35: Run Time Parameters for ESN iteration 06-01 V15.

Description	**This Iteration**	Justification	02-01 Iteration
Input values	3 vectors of data	Input choice	3 vectors of data
Classes	Mount only	Class choice	Mount only
Number of neurons	336	From 02-01	336
Input scaling	0.9252	From 02-01	0.9252
Neuron leak rate	0.1379	From 02-01	0.1379
Regularisation	0.0001	From 02-01	0.0001
Spectral Radius	0.9455	From 02-01	0.9455

6.10.8 *Training Results (06-01 V15)*

The training error for this iteration was more than two and a half times as large (see table 6.36) as that from the initial model. Training completed in 14.35 seconds elapsed, this is much more computational resources than the initial 02-01 model. The additional computational requirements are as a result of using the larger training file.

Table 6.36: Training Results for ESN iteration 06-01 V15.

Description	**This Iteration**	02-01 Iteration
User Time in seconds	**14.19**	7.28
System Time in seconds	**0.16**	0.09
Elapsed Time in seconds	**14.35**	7.50

6.10.9 *Testing Results (06-01 V15)*

Table 6.37: Processing Times for Concatenated Test Datasets iteration 06-01 V15.

Iteration Id	**06-01**	02-01
User Time in seconds	30.96	29.94
System Time in seconds	00.97	00.89
Elapsed Time in seconds	32.01	30.83

Table 6.38: Classification Metrics for Concatenated Test Datasets iteration 06-01 V15.

Iteration Id	**06-01**	02-01
ESN Test Error	0.3349	0.2046
RMSE	0.0323	0.0874
AUC	0.9695	0.9418

Table 6.39: Results for Concatenated Test Datasets iteration 06-01 V15.

Measures at 0.6 Cut-Off				
Description	**06-01**		02-01	
Classification rate	0.286		0.714	
	Base Cl	**Mount Cl**	Base Cl	Mount Cl
Precision	0.9995	0.1220	0.9996	0.0737
Recall	0.9998	0.0442	0.9981	0.2920
Confusion Matrix at 0.6 Cut-Off				
	Predicted 06-01		Predicted 02-01	
Actual	**Base Cl**	**Mount Cl**	Base Cl	Mount Cl
Base Cl	217674	36	217295	415
Mount Cl	108	5	80	33

The area under the ROCcurve varied (0.8872,0.9694 and 0.8458) across the three filtered iterations with this iteration (V15) having the largest area and this is significantly more than the area under the ROC curve for the original, 02-01 iteration. However, the classification rate at the 0.6 cut-off level has fallen (Table 6.39) substantially compared with iteration 02-01 and is the same as iteration 05-01 V13. Looking at the ROC curve in Figure 6.30a, this ESN model is more discriminative at low model output levels (less than 0.2) and about the same from 0.2 and above.

Mount precision at the selected cut-off level has increased by a noticeable amount, while Mount Recall has fallen with this model. There has been a substantial drop in False Positives at the 0.6 cut–off (36 versus 415) but that is juxtaposed against a drop in both True Positives (5 versus 33) and in the Classification Rate (0.286 versus 0.714).

Looking at Figure 6.29 and comparing it with the equivalent figure for the initial 02-01 iteration (Figure 6.5 on page 155), the ESN output for this iteration is considerably more subdued than 02-01 with much less variation and that this accounts for both the drop in false positives and the drop in true positives. The ESN model response for iteration 06-01 V15 has tended to cluster around zero.

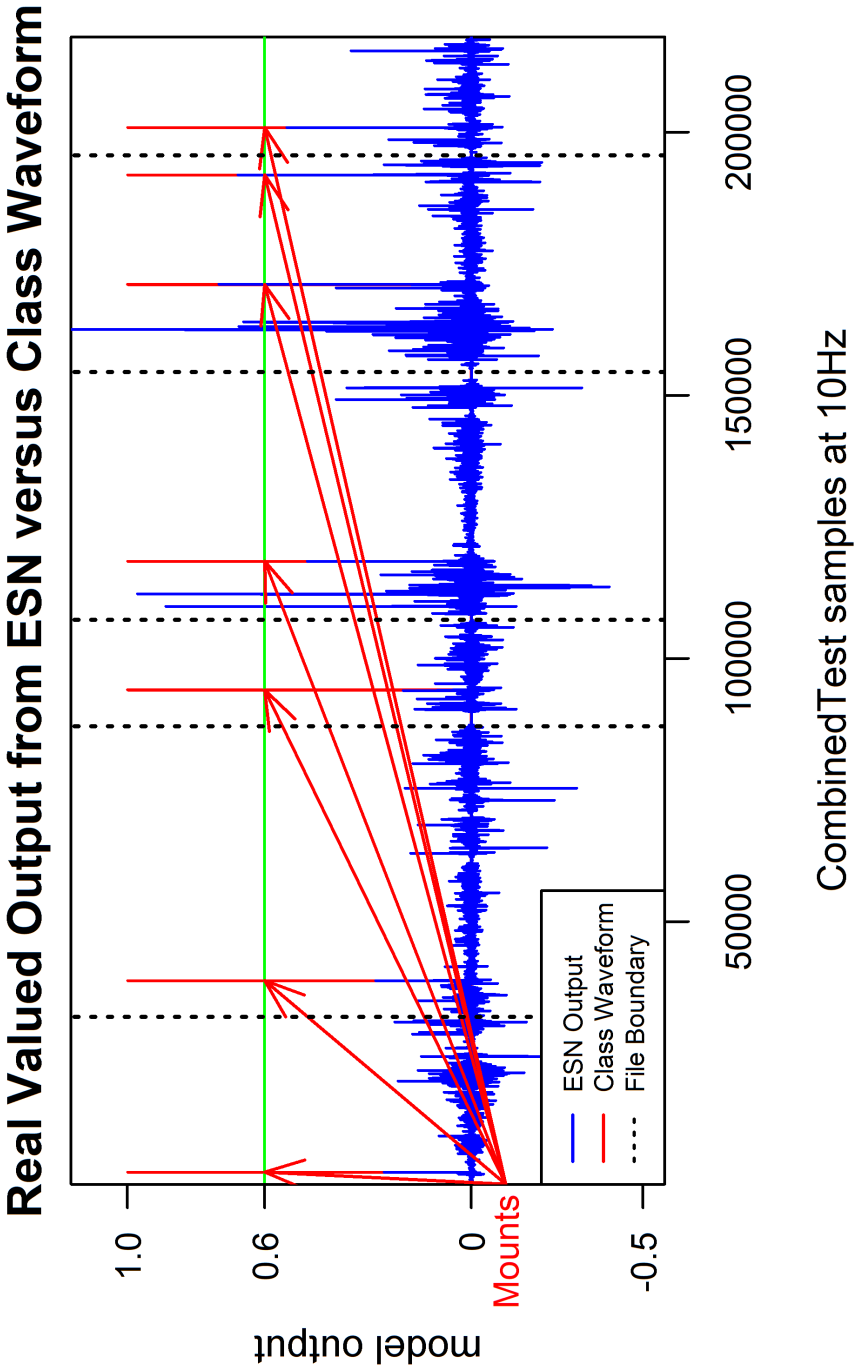
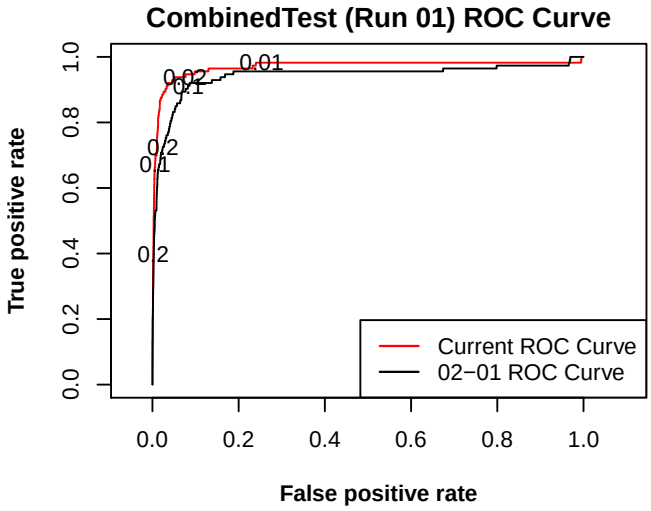


Figure 6.29: 06-01 V15 ESN Output for Concatenated Test Data



(a) ROC Curve

Figure 6.30: 06-01 V15 Performance Plots for Concatenated Test Data

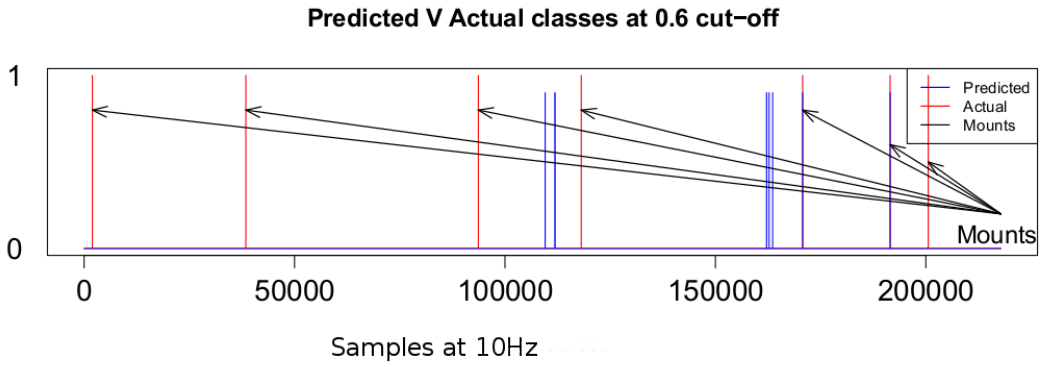


Figure 6.31: 06-01 V15 Classes for Concatenated Test Data

6.10.10 *Results versus Goals (06-01) V15*

This iteration met two of its design goals but did not meet the third. The area under the ROC curve for this design iteration is greater than the goal, the number of False Positive has dropped substantially but the classification rate is less than the goal.

6.10.11 *Discussion (06-01) V15*

This design iteration has produced better results than the initial ESN model in two areas but suffers from a substantially lower classification rate. There is a sense that progress towards the overall goal of this research is being made and future iterations will incorporate a Butterworth filter to detrend the Gyroscope data along with a rolling mean to smooth the input including any outliers.

6.11 ACCELEROMETER - ADDITIVE #2 07-01 V23

Net acceleration power is a relatively common feature used within IMU based activity classification although different authors call it by different names. For example, Z. He, Liu, Jin, Zhen, and Huang (2008) call this feature "*net acceleration independent of orientation*" while Kwapisz, Weiss, and Moore (2011) call this feature "*average resultant acceleration*". Net acceleration power was chosen because its value is simple to calculate and it is independent of the sensor orientation, as reported by Z. He et al. (2008) and Kwapisz et al. (2011). With the sensor in this research being wrist mounted the orientation of the sensor changes often, especially during mounting and as explained in the Data section (Section 4.6), a change in the orientation of the sensor causes changes across the three axis of the accelerometer and so having an orientation independent value makes classification more probable. Experience gained during the exploration phase confirmed that using the three separate acceleration axis was less useful than using net acceleration power.

The aim of this set of iterations was to build an ESN using data which includes both the three axis of linearly transformed gyroscope data and the "net acceleration power" from the accelerometer data. The net acceleration power was calculated by summing the squares of each axis then taking the square root of the product. The resultant number was then scaled to be between 0 and 1. The goal in this iterative step was again to build and test a model using modified input data that includes the additional accelerometer signal and then measure that model against the initial 02-01 model. As with recent prior work this set of design iterations uses a static set of ESN model parameters copied from iteration 02-01, as noted in section 6.8.12. Six different design iterations were run and tested, identified as V21 through V27 (excluding V24), and the best performing iteration, V23, is described in detail with any important factors from the other iterations noted in passing.

The first iteration in this series (V21) used the net acceleration power and obtained a somewhat disappointing area under the ROC curve of 0.91997. Summing the square of three numbers with values ranging between zero and one and then taking the square root of that product gives a value between zero and 1.732 while the three Gyroscope vector range between zero and one. The ESN activation function, $tanh$, has the affect of non-linearly scaling its output values so that they are between zero and one. This means that input values greater than one have very little influence on the ESN model and this may account for the disappointing result. For the second iteration (V22), the net acceleration power was divided by 2 to ensure that all input lay between zero and one and this produced a more useful result with the area under the ROC curve of 0.96254. The V23 iteration used a Butterworth filter to split the net acceleration power into a low frequency and a higher frequency band at 5Hz. 5Hz was chosen as it was suggested as an upper bound for walking by Bouten, Koekkoek, Verduin, Kodde, and Janssen (1997) and because it was conveniently half of the data sample rate. This iteration produced very encouraging results with the area under the ROC curve of 0.97315 and the result from this iteration were the best from this set of iterations.

Three further design iterations were attempted in this series, for iteration V25 the input data was processed exactly the same as for iteration V23, however an ESN

model with 1,000 neurons was built (much larger neuron number) and the other model parameters were taken from a recommendation by Lukoševičius in (Lukoševičius, 2012). This produced less interesting results with the area under the ROC curve of 0.94935. Design iteration V26 was similar except that an ESN model with 1,200 neurons was built and its results were even worse with the area under the ROC curve of 0.93173. For the last design iteration in this series, V27, the ESN parameters were reverted back to those used in 02-01 except that the number of neurons was raised to 500 (from 336). The results from this iteration were also disappointing with the area under the ROC curve of 0.94269.

Please note that this is not a comprehensive set of design iterations to provide insight into the best possible way to incorporate acceleration data into ESN classifiers as the large scale of such a goal is outside of this research but is instead a way to try a limited set of ideas relating to incorporating acceleration data.

6.11.1 *Problem Identification (07-01)*

Earlier instantiations of the ESN classification engine have demonstrated a good ability to classify *mounts* based on real-world data but while these initial results are quite impressive they are not "good enough" to use in real life. Earlier instantiations have thrown out too many False Positives and have not successfully classified all of the mounts. Essentially the discriminative ability of the ESN classification engine needs to be improved.

6.11.2 *Motivation (07-01)*

The motivation for this design iteration was to explore the addition of net acceleration power to the Gyroscope data as an aid to improving the ability of the ESN classification engine to discriminate the activity of interest.

6.11.3 *Design Goals (07-01 - V23)*

The goals for this design iteration are:

1. Maintain or improve the classification rate from 0.714, as achieved with iteration 02-01.
2. Improve the area under the ROC curve from 0.9694, as achieved with iteration 06-01 V15.
3. Maintain or reduce the number of false positives from 36, as achieved with iteration 06-01 V15.

6.11.4 *Measure of Success (07-01 - V23)*

An overall improvement in the area under the ROC curve while maintaining or improving the classification rate as defined in section 6.5.10 and either maintaining or reducing the number of False Positives.

6.11.5 *Data Description (07-01 - V23)*

Training Data; concatenation of:

- Partial 0813-1 LA (laboratory) – Subset from end including two mounts
- Partial 0716-2 RA – Central (riding) region reduced
- Partial 0912-1 RF – Central (riding) region reduced
- Partial 0912-2 RG – Central (riding) region reduced
- Partial 0913-3 RF – Central (riding) region reduced

Data from the riding phase was removed from each dataset in larger amounts compared with the initial 02-01 model. Only two sections of 1,000 samples of riding phase data were included from each dataset, 1,000 from immediately after the mount and 1,000 immediately preceding the dismount.

Testing Data; concatenation of:

- Full 0719-1 RB
- Full 0812-2 RA
- Full 0829-1 RC
- Full 0830-3 RD
- Full 0902-1 RE
- Full 0902-2 RC

The entire data file was always used for testing files.

6.11.6 *Static Parameter Description (07-01 - V23)*

The following static parameters were used during this iteration:

- Error Cost Function: (conditional mean square error class + conditional mean square error non-class)$/2$
- Included Sensors: Gyroscope and Accelerometer
- The Gyroscope and Accelerometer sensor readings are linearly transformed, real-values ranging between 0 and 1.
- The Gyroscope data was passed through a second order high-pass Butterworth filter with a frequency of 0.01Hz, designed to remove signal drift.

- The detrended Gyroscope data was then transformed with a rolling mean.
- No outliers were removed from either sensor, however the rolling mean applied to the Gyroscope data would tend to dampen the Gyroscope outliers.
- The three axis of Accelerometer data are combined into a single vector of values by summing the squares of each axis then taking the square root of the product.
- The single Accelerometer vector is then transformed into two vectors by using a Butterworth filter to separate the signal into upper and lower frequency bands.

6.11.7 *Run time Parameter Description (07-01 V23)*

The parameters from iteration 02-01 were used for this design iteration and are shown in table 6.40 and the ESN model that was built for this set of design iterations was built, trained and tested on a Windows 7 based PC running an i7 processor and 8GB RAM.

Table 6.40: Run Time Parameters for ESN iteration 07-01 V23.

Description	**This Iteration**	Justification	02-01 Iteration
Input values	5 vectors of data	Input choice	3 vectors of data
Classes	Mount only	Class choice	Mount only
Number of neurons	336	From 02-01	336
Input scaling	0.9252	From 02-01	0.9252
Neuron leak rate	0.1379	From 02-01	0.1379
Regularisation	0.0001	From 02-01	0.0001
Spectral Radius	0.9455	From 02-01	0.9455

6.11.8 *Training Results (07-01 V23)*

The training error for this iteration was more than two and a half times as large (see table 6.41) as that from the initial model. Training completed in 14.2 seconds elapsed, this is much more computational resources than the initial 02-01 model. The additional computational requirements are as a result of using the larger training file.

Table 6.41: Training Results for ESN iteration 07-01 V23.

Description	**This Iteration**	02-01 Iteration
User Time in seconds	**14.03**	7.28
System Time in seconds	**0.16**	0.09
Elapsed Time in seconds	**14.20**	7.50

Table 6.42: Processing Times for Concatenated Test Datasets iteration 07-01 V23.

Iteration Id	**07-01**	02-01
User Time in seconds	31.47	29.94
System Time in seconds	01.11	00.89
Elapsed Time in seconds	32.64	30.83

Table 6.43: Classification Metrics for Concatenated Test Datasets iteration 07-01 V23.

Iteration Id	**07-01**	02-01
ESN Test Error	0.2373	0.2046
RMSE	0.0299	0.0874
AUC	0.9732	0.9418

Table 6.44: Results for Concatenated Test Datasets iteration 07-01 V23.

Measures at 0.6 Cut-Off				
Description	**07-01**		02-01	
Classification rate	0.571		0.714	
	Base Cl	**Mount Cl**	Base Cl	Mount Cl
Precision	0.9996	0.8462	0.9996	0.0737
Recall	0.9999	0.1947	0.9981	0.2920
Confusion Matrix at 0.6 Cut-Off				
	Predicted 07-01		Predicted 02-01	
Actual	**Base Cl**	**Mount Cl**	Base Cl	Mount Cl
Base Cl	217706	4	217295	415
Mount Cl	91	22	80	33

6.11.9 *Testing Results (07-01 V23)*

All three measures of success have improved substantially from the immediately prior iteration, the area under the ROC curve has increased from 0.9694 to 0.97315 (Table 6.44), the number of false positives has dropped from 36 to 4 and the classification rate has increased from 0.286 to 0.571. Both the area under the ROC curve and the number of False Positives has meet the design goals, however, while the classification rate has improved it has not met the design goal and is still less than than the 0.714 achieved with iteration 02-01. These figures are summarised in Table 6.44. Looking at the ROC curve in Figure 6.34a, this ESN model is more discriminative at all model output levels less than 0.2 and has a similar discriminative level above that. The other iterations for this set produced the following area under the ROCcurve V21 – 0.9200, V22 – 0.9625, V25 0.9494, V26 – 0.9317 and V27 – 0.9427.

Mount precision at the selected cut-off level and across the board has increased by a noticeable amount and Mount Recall has also increased substantially with this model, this is demonstrated in Figure 6.34b. There has been a further substantial drop in False Positives at the 0.6 cut–off (4 versus 415) and in addition the number of True

Positives at the 0.6 cut-off level has increased from the last iteration (06-01) but is still short of the 33 achieved during 02-01. The Classification Rate at 0.571 has also increased from 06-01 but is again short of that achieved during 02-01 at 0.714).

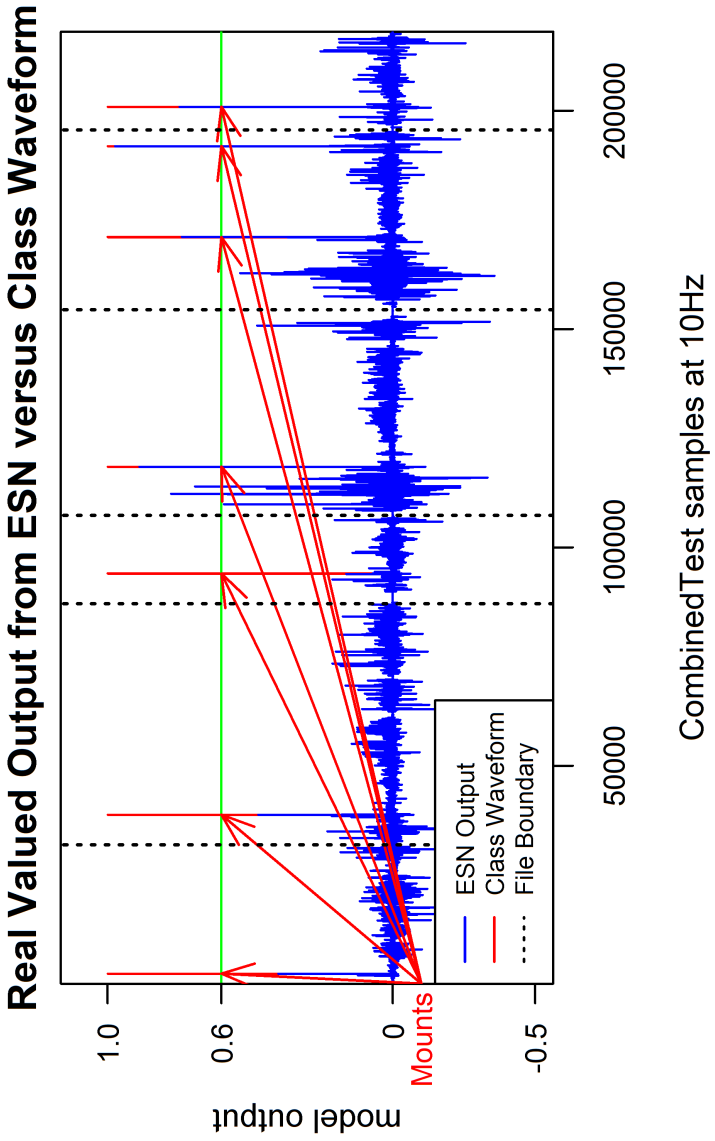
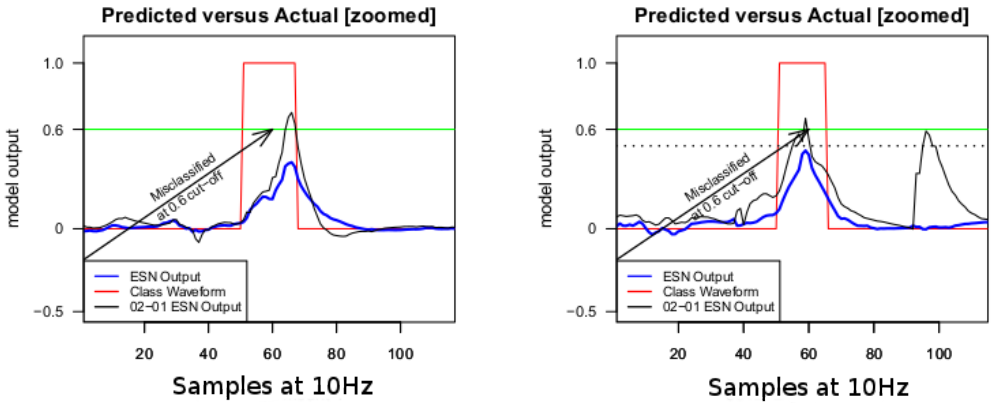


Figure 6.32: 07-01 V23 ESN Output for Concatenated Test Data

Looking at Figure 6.32 and comparing it with the equivalent figure for the initial 02-01 iteration (Figure 6.5 on page 155), the ESN output for this iteration is considerably more subdued than 02-01 overall with much less variation but the output from th eESN does rise substantially for all mounts except the third mount. Even the first two mounts which did not reach the 0.6 cut-off produce a substantial rise in the ESN output (see zoomed in mount Figures 6.33a and b). Together this accounts for

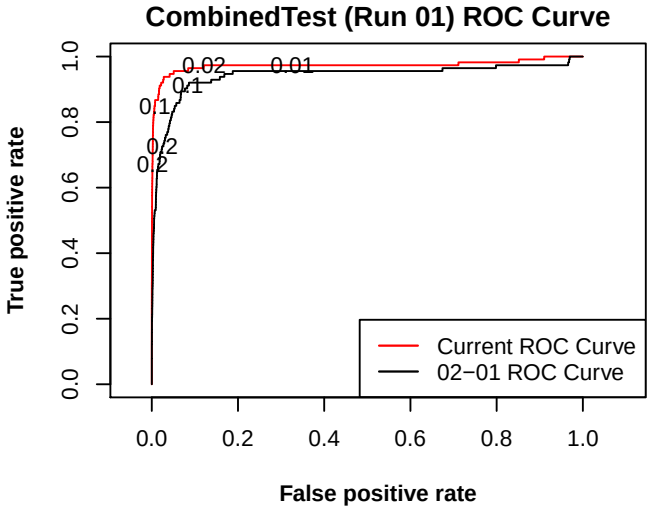
both the drop in false positives and the rise in true positives. The False Positives that reach the 0.6 cut-off are in two clusters, one just prior to the 4th mount and the other just prior to the 5th mount. On re-checking the video for these two mounts it was apparent that both clusters coincide with brushing the horse prior to tacking up.

Out of curiosity and while the video files were loaded to check the clusters of False Positives, a decision was made to also look at mount 0829-2 RC. This mount has been consistently misclassified across all ESN models (see Figure 6.33c for an example from this iteration). Somewhat embarrassingly, the video shows that this mount was not a stirrup mount like all the other mounts in the training and testing data. Instead it is a mount from a mounting block. Mounting blocks raise the rider's height compared with the horse and so an assisted mount from a mounting block results in considerably different body movements from the rider, especially different hand movements. In particular, with a mounting block there is no sudden elevation of the rider's forearm as they mount.

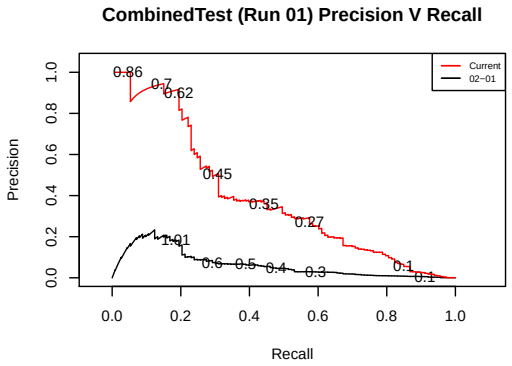


(a) 0719-1 RB #1 (b) 0812-2 RA #1

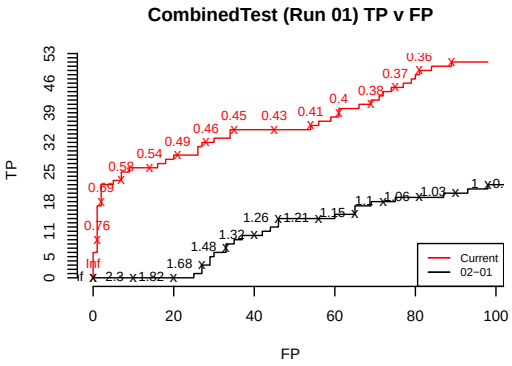
Figure 6.33: 07-01 V23 First Two Mounts for Concatenated



(a) ROC Curve



(b) Precision V Recall



(c) TP V FP

Figure 6.34: 07-01 V23 Performance Plots for Concatenated Test Data

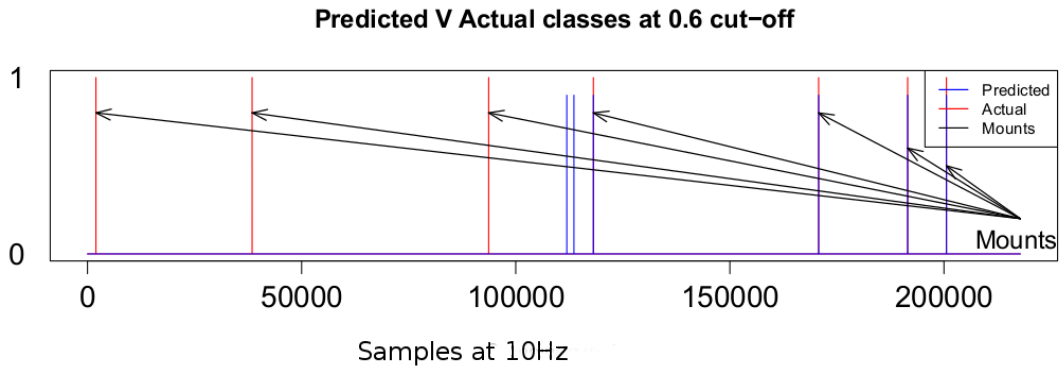


Figure 6.35: 07-01 V23 Classes for Concatenated Test Data

6.11.10 *Testing Results Selected Summary (07-01 V25–V27)*

This section summarises some of the more significant results from design iterations V25 through V27. While design iteration 07-01 V23 produced the highest area under the ROC curve, other design iterations produced lower areas under the ROC curve, but still reasonably high whilst producing a better classification ratio or less False Positives and so these results are summarised below. The equivalent results for the initial 02-01 iteration are also included for comparative purposes.

Table 6.45: Comparative results for 07-01 V23–V27.

	02-01	**V23**	**V25**	**V26**	**V27**
ESN Parameters					
Model Set	02-01	02-01	Lukosevicius	Lukosevicius	02-01
Num. Neurons	336	336	1,000	1,200	500
Overall Classification Metrics					
RMSE	0.087	0.030	0.027	0.027	0.035
AUC	0.942	0.973	0.949	0.932	0.943
Classification Metrics at 0.6 Cut-off					
Rate	0.714	0.571	0.571	0.571	0.714
Mount Precision	0.074	0.846	0.909	1.000	0.562
Mount Recall	0.292	0.195	0.089	0.080	0.283
False Positives	376.000	4.000	1.000	0.000	25.000
True Positives	40.000	22.000	10.000	9.000	32.000
Classification Metrics at 0.39 Cut-off					
Rate	0.714	0.857	0.857	0.857	0.857
Mount Precision	0.053	0.383	0.809	0.786	0.266
Mount Recall	0.504	0.363	0.336	0.292	0.407
False Positives	1012.000	66.000	9.000	10.000	127.000
True Positives	57.000	41.000	38.000	33.000	46.000

6.11.10.1 *Design Iteration 07-01 V25*

This design iteration used the same input data as V23 but substituted ESN model parameters suggested by Lukoševičius in (Lukoševičius, 2012) with 1,000 neurons versus 336 neurons for V23, key metrics from this iteration are summarised in Table 6.45 on page 214.

This design iteration produced an output that was slightly more subdued than the output from V23 overall, as noted by the lower Root Mean Square Error (RMSE), Figure 6.36 on page 215 provides a visual indicator of this. This iteration classified the mount activities with more precision at the selected 0.6 cut-off (0.9091 V 0.8462). In addition it only produced a single False Positive at that cut-off point versus 4 for V23. This design iteration produced an area under the ROC curve of 0.94935 that demonstrates that it was somewhat less discriminative at very low levels (under around 0.025) but similarly discriminative at levels above that (Figure 6.37 a, page 216. The classification rate for this design iteration of 0.571 at the 0.6 cut-off

was equal to the classification rate from V23 at the same cut-off point. At a 0.39 cut-off level this design iteration also bests the initial 02-01 results. A 0.39 cut-off level produces 9 False Positives, 38 True Positives, a precision rate of 0.8085 and a recall rate of 0.3362. The precision at this level is much higher than V23 but the recall rate is slightly less.

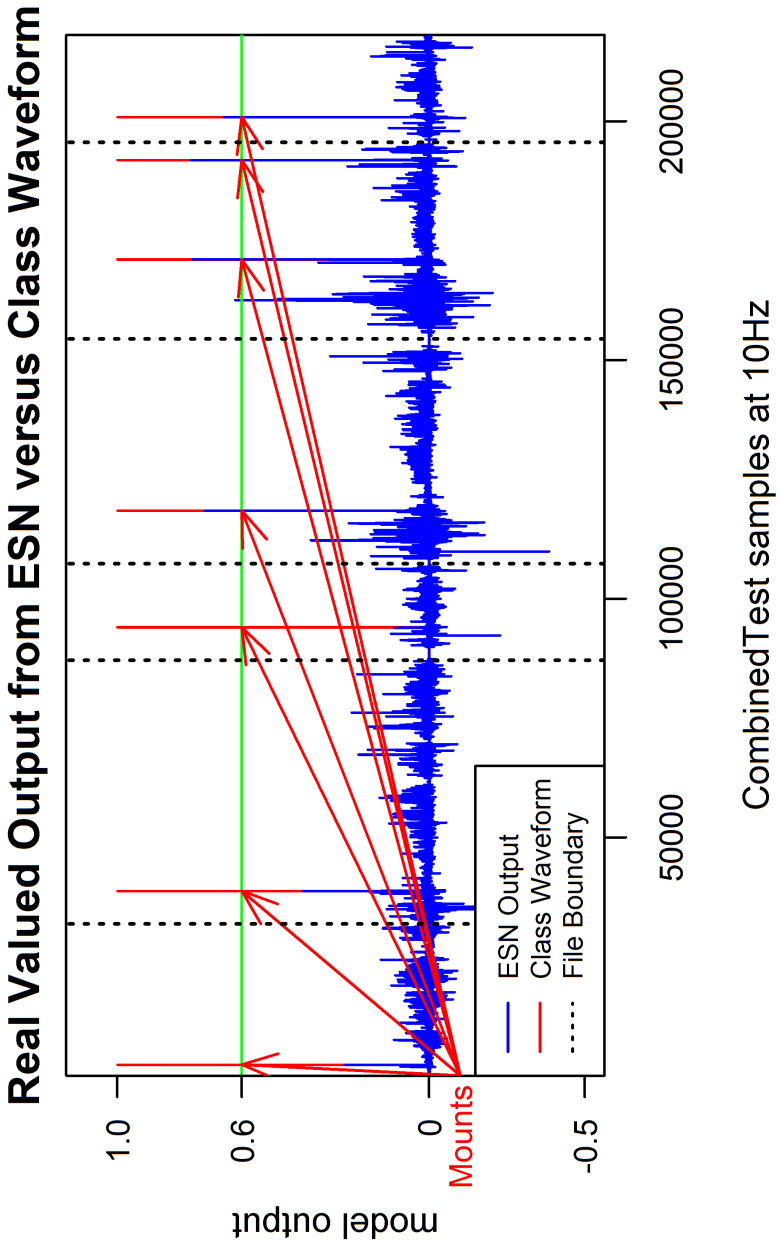
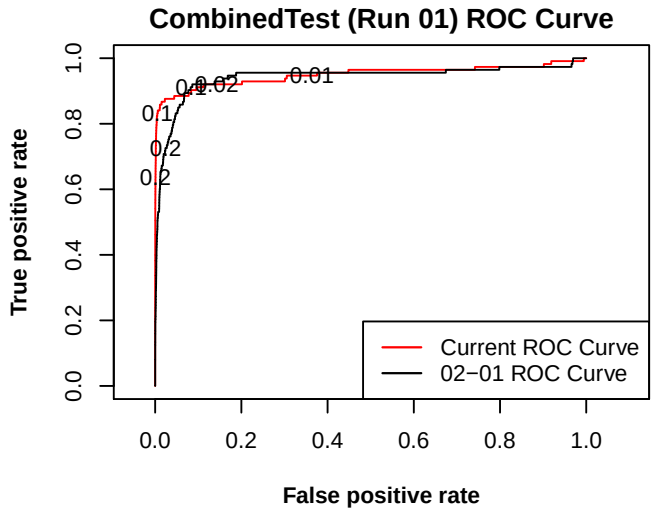
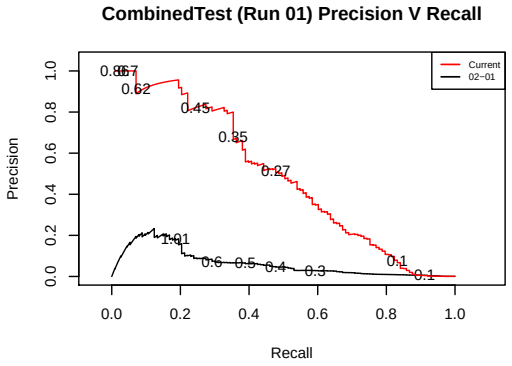


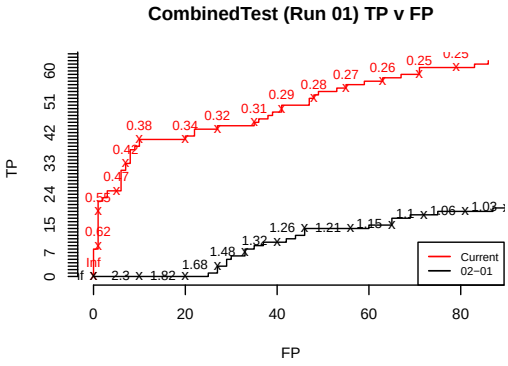
Figure 6.36: 07-01 V25 ESN Output for Concatenated Test Data



(a) ROC Curve



(b) Precision V Recall



(c) TP V FP

Figure 6.37: 07-01 V25 Performance Plots for Concatenated Test Data

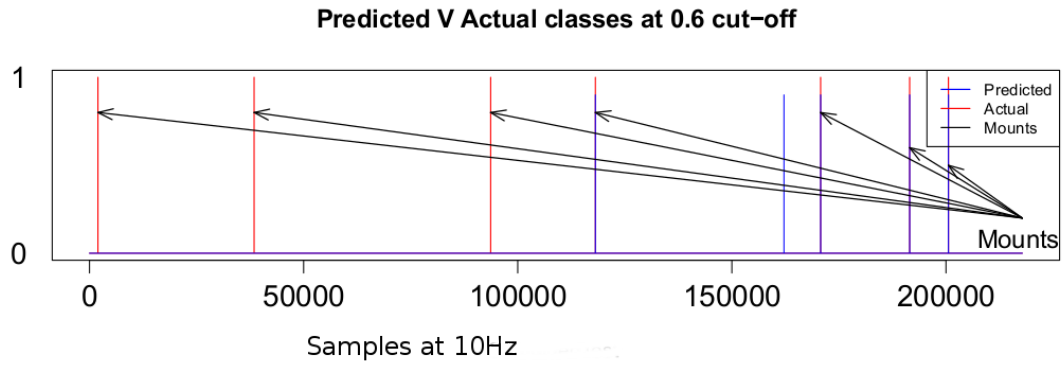


Figure 6.38: 07-01 V25 Classes for Concatenated Test Data

6.11.10.2 *Design Iteration 07-01 V26*

This design iteration used the same input data as V23 but substituted ESN model parameters suggested by Lukoševičius in (Lukoševičius, 2012) with **1,200** neurons and again key metrics from this iteration are summarised in Table 6.45 on page 214.

This design iteration produced an output that was slightly more subdued than the output from V23 and V25 overall, as noted by the lower RMSE, Figure 6.39 on page 217 provides a visual indicator of this. In addition, this iteration managed to correctly classify all the base class samples at the selected 0.6 cut-off and as a result classified the mount activities with more precision at that level (1.0 V 0.84615) and no False Positives at that cut-off point. However at the 0.6 cut-off level the Mount recall for this iteration was much less (0.07965 V 0.19469).

This design iteration produced an area under the ROC curve at 0.93173 that demonstrates that it was somewhat less discriminative at very low levels (under around 0.025) compared with V23, but similarly discriminative at levels above that (Figure 6.40 a, page 218). The classification rate for this design iteration of 0.571 at the 0.6 cut-off was equal to the classification rate from V23 at the same cut-off point. At a 0.39 cut-off level this design iteration also bests the initial 02-01 results. A 0.39 cut-off level produces 9 False Positives, 38 True Positives, a precision rate of 0.8085 and a recall rate of 0.3362.

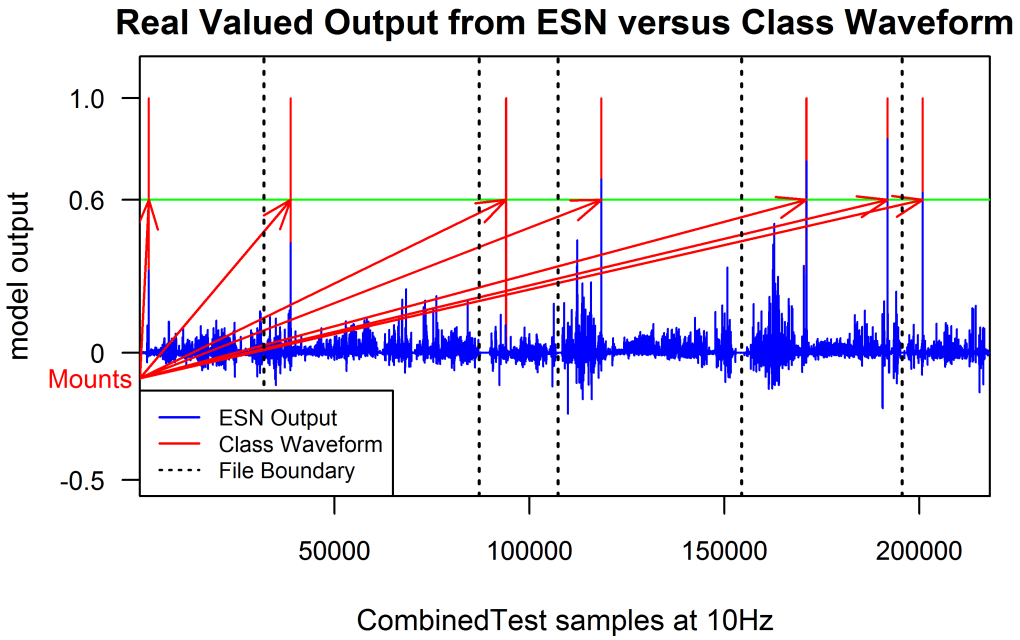
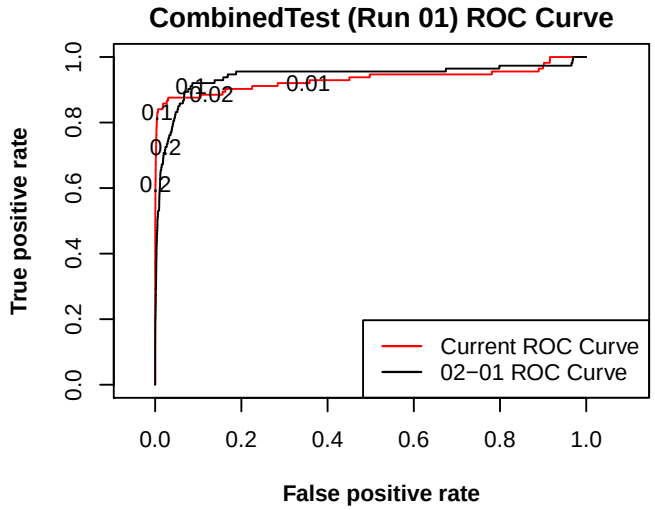
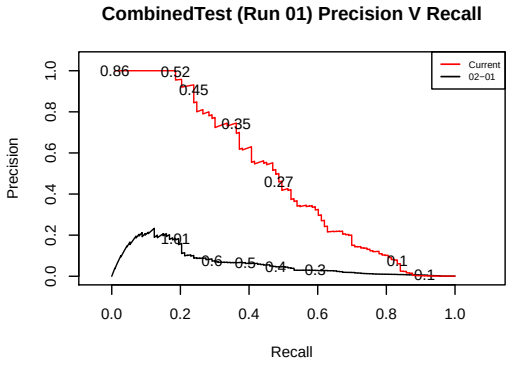


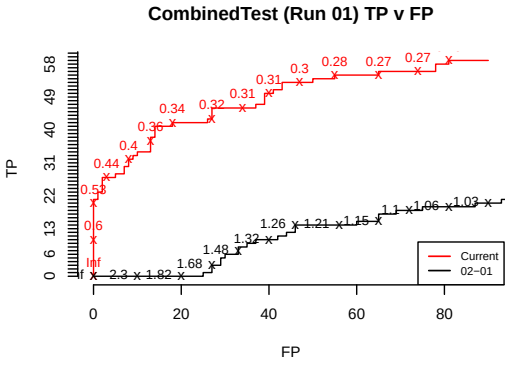
Figure 6.39: 07-01 V26 ESN Output for Concatenated Test Data



(a) ROC Curve



(b) Precision V Recall



(c) TP V FP

Figure 6.40: 07-01 V26 Performance Plots for Concatenated Test Data

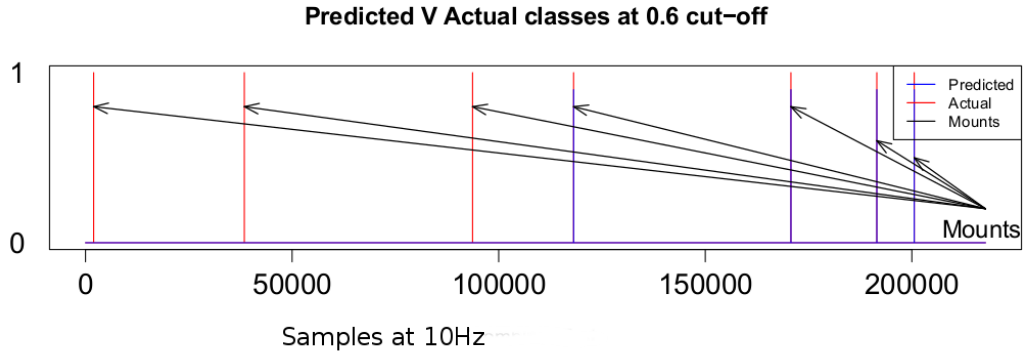


Figure 6.41: 07-01 V26 Classes for Concatenated Test Data at 0.6 Cut-off

6.11.10.3 *Design Iteration 07-01 V27*

This design iteration used the same input data as V23 along with the 02-01 ESN model parameters but with **500** neurons and again key metrics from this iteration are summarised in Table 6.45 on page 214. Interestingly, the outputs from this model meet all of the design goals for improving on the metrics from 02-01. The classification rate is the same, the area under the ROC curve is greater and there are less False Positives, however, this iteration was not chosen as the best in set.

This design iteration produced an output that demonstrated more energetic responses than the output from V23, V25 and V26 overall but a more subdued response than that from 02-01, as noted by the higher RMSE, Figure 6.42 on page 219 provides a visual indicator of this. However, the more energetic responses have resulted in more False Positives and while more mounts were classified, this was done with fewer True Positives. Effectively this design iteration traded off a lower precision (0.5614) in return for a better recall (0.2832).

This design iteration produced an area under the ROC curve at 0.94269 that demonstrates that it was somewhat less discriminative at very low levels (under around 0.025) compared with V23, but similarly discriminative at levels above that (Figure 6.43 a, page 220). The classification rate for this design iteration of 0.714 at the 0.6 cut-off was equal to the classification rate from 02-01 and better than V23 at the same cut-off point. At a 0.39 cut-off level this design iteration also bests the initial 02-01 results in all areas except the number of True Positives although it has the next highest number of these. A 0.39 cut-off level produces 127 False Positives, 46 True Positives, a precision rate of 0.2659 and a recall rate of 0.4071.

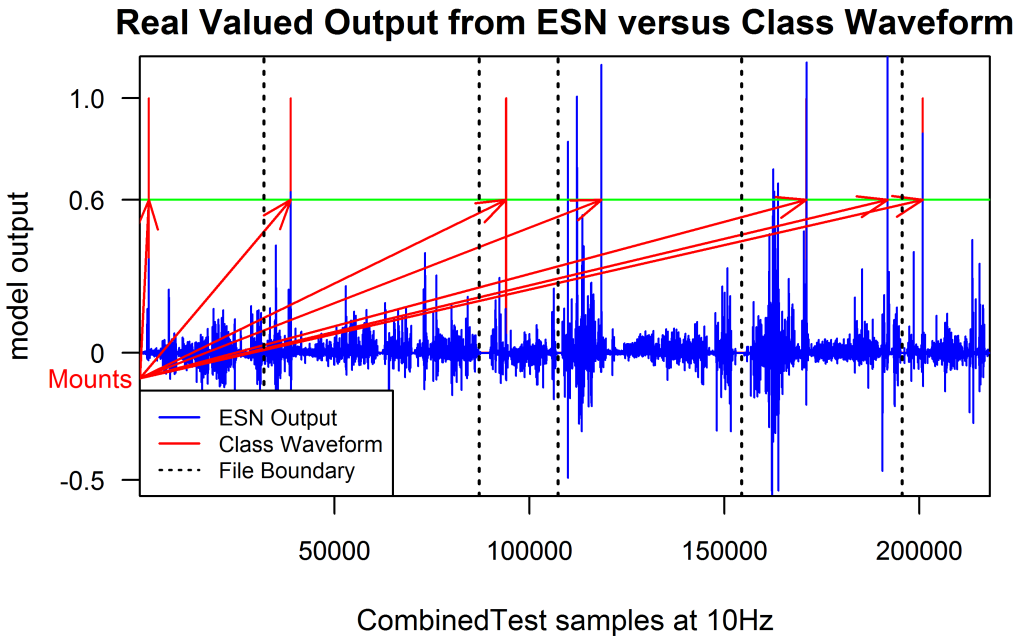
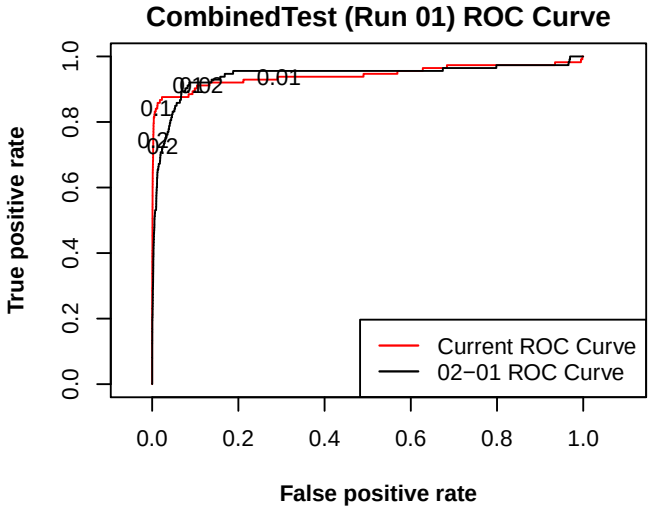
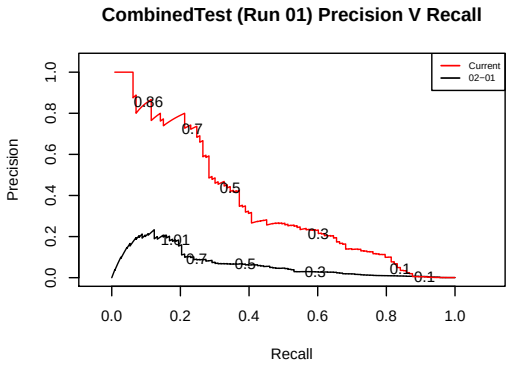


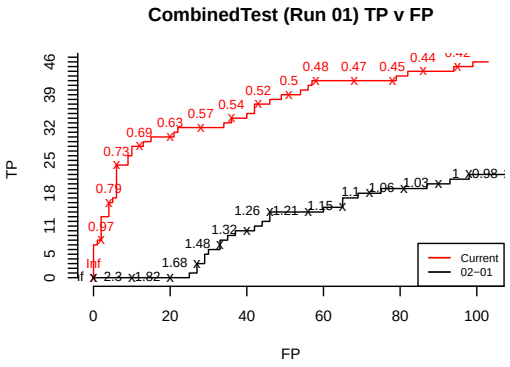
Figure 6.42: 07-01 V27 ESN Output for Concatenated Test Data



(a) ROC Curve



(b) Precision V Recall



(c) TP V FP

Figure 6.43: 07-01 V27 Performance Plots for Concatenated Test Data

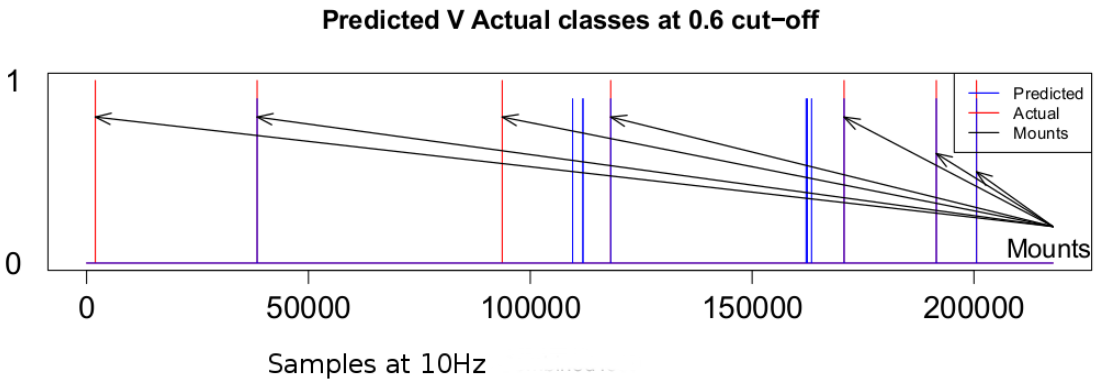


Figure 6.44: 07-01 V27 Classes for Concatenated Test Data at 0.6 Cut-off

6.11.11 *Results versus Goals (07-01) V23*

The selected model from this design iteration set met two of its design goals but did not meet the third. The area under the ROC curve for this design iteration is greater than the goal and the number of False Positives at a 0.6 cut-off level has dropped substantially but the classification rate at the same 0.6 cut-off level is less than the goal. A different model (V27) did meet all three design goals but it was not the selected model because it had a much higher level of False Positives. Yet another iteration from this set (V25) had very similar but better metrics to V23 in all areas except the area under the ROC curve. All four design iterations produced pleasing results and this supports the value of including the Accelerometer data in the input data.

6.11.12 *Discussion (07-01) V23 and other iterations*

The selected design iteration (V23) has produced better results than the initial ESN model in two areas but suffers from a lower classification rate at the 0.6 cut-off level. In many ways the output response from this ESN model is more desirable than the output response from the 02-01 model. Looking at Figure 6.35, for example, which shows the model output after the 0.6 cut off has been applied and comparing it with Figure 6.8 on page 157, it can be seen that there are almost no False Positives (two shown between mounts three and four) and four of the seven mounts are clearly defined. Then, taking into account that the third mount is correctly misclassified (as it is not a stirrup mount) it looks even better with two thirds of the six stirrup mounts being correctly classified. Lastly, the cut-off level of 0.6 was chosen towards the start of this research in a somewhat arbitrary manner based on gut feel of a desired response level from a classifier. There is no other logic associated with this cut-off level. This cut-off level was maintained through each design iteration for consistency but now that there are more examples of potential output levels from a "useful" ESN classification model it seems logical to review this cut-off level.

At a 0.39 cut-off level, for example, this design iteration would correctly classify all six of the stirrup mounts, resulting in a classification rate of 0.857 (1.0 based on removing the assisted mount) and an associated rise in False Positives from a total of four samples at a 0.6 cut-off to a total of 66 samples at 0.39. Of course, in order to compare like with like the 02-01 cut-off level would also need to be lowered to 0.39 and in this case the 02-01 design iteration would misclassify one stirrup mount, the last one and even more tellingly, the number of False Positives produced from 02-01would rise substantially to 1,012.

The output from the current design iteration is more subdued but more discriminative. Based on the proposed new cut-off level, this model with the addition of the additional information in the form of the net acceleration power substantially improves the classification of the Stirrup Mount activity using an ESN classification model. The improvement in classification is such that this model can be considered good enough to demonstrate that the overall classification goal for this research had been achieved.

Design iteration V25 (Subsection 6.11.10.1) also bested iteration 02-01 at a 0.39 cut-off, with a similar more subdued and more discriminative output, although not quite as good as V23. This supports a conclusion that this combination of input data features provides good classification across a variety of different ESN model characteristics and supports the conclusion that a robust artefact has been produced that meets the research objectives.

6.11.13 *Next Step*

At this stage in the research process the project was considerably beyond the anticipated finish time, the funding scholarship had ended and both academic and home pressures were strongly pushing for an end to the Design Cycle phase so that the thesis could be completed and submitted for examination. There was only time for one more design iteration. A decision was made to further test the robustness of the ability of the ESN model to successfully classify punctual activities by substituting a different version of the R ESN code library.

All of the classification within this research that is based on ESNs use a library that implements the ESN concept as described in a paper authored by Lukoševičius and Jaeger (Lukoševičius and Jaeger (2009)), where the R library code was initially written by Schliebs and made available for this research. One of Schliebs objective in writing the library code was to produce efficient R code that was simple to run on parallel processors so that any search across the model parameter set would use the minimum of resources. In meeting this goal Schliebs replicated the essential and core function associated with ESN models as expressed in Lukoševičius and Jaeger (2009) but did not implement some of the "nice-to-have" features such as the automatic incorporation of a vector of ones within the input data, as recommended (for example) by Lukoševičius in (Lukoševičius, 2012). The library was then subsequently modified by the author over a number of iterations but the basic structure of the code was maintained.

During earlier work the author came across a website where Lukoševičius had published a small number of sample code files for implementing a minimal ESN in various programming languages, including R. This was at http://minds.jacobs-university.de/mantas/code and was cited in conjunction with his publication Lukoševičius (2012). For some reason Lukoševičius withdrew the original page (URL) where the code was hosted but then subsequently reposted the code at http://212.201.49.24/mantasCode and http://212.201.49.24/sites/default/files/uploads/mantas/code/minimalESN.R. During the exploration phase of this research the author downloaded and tested this code in a limited manner, found a coding error and corrected that.

A decision was made to use the remaining time to demonstrate the robustness of the ESN model for this work by repeating the selected iteration (V23) from last set of design iterations using exactly the same data and ESN parameters from the V23 iteration but substituting the Lukoševičius R code for the Schliebs ESN R code library. The Lukoševičius R code looked like it had been ported from a MatLab or some other implementation as it used very simple R statements and did not attempt to use any R specific statements that would have improved its efficiency. In addition, this

alternate code implemented two ideas that were not implemented within the Schliebs code. The first difference was that the alternate code automatically truncated the early output from the ESN during training (only) because Lukoševičius and Jaeger had discovered that during training the ESN model tended to produce unstable output as the initial data was parsed and until the internal "memory" was filled up. The second, and probably more crucial, change in the code libraries was the implementation of the additional fixed value input data vector.

The next and last design iteration, described in the following section, was designed to demonstrate that an alternate code base with a slightly different implementation of the ESN model within R would produce a very similar response from the same data.

6.12 LUKOSEVICIUS CODE 08-01 V28

The aim of this iteration was to build an ESN using the same data used by iteration 07-01 V23 along with the same ESN parameters used by 07-01 V23 so that the classification robustness of ESN models classifying a punctual activity could be demonstrated. In this case, however, a decision was made to increase the number of neurons from 336 to 1,000. Lukoševičius tended to use 1,000 in his ESN models and as this iteration used his code a decision was made to also use his recommended number of neurons.

6.12.1 *Problem Identification (08-01)*

Design iteration 07-01 demonstrated a very good ability to classify *mounts* based on real-world data including a "good enough" reduction in False Positives to enable a ESN based classifier to be used in real life. As discussed in Section 6.11.13 on page 222, this classifier and dataset have not yet been subject to external review and so some critics may question the robustness of this solution.

6.12.2 *Motivation (08-01)*

The motivation for this design iteration was to partially demonstrate the robustness of the code and dataset used to classify *mounts* by substituting an alternate ESN code library while retaining the same dataset.

6.12.3 *Design Goals (08-01 - V28)*

The goals for this design iteration are:

1. Achieve a similar or better classification rate to 07-01 V23 (0.571).
2. Achieve a similar or better area under the ROC curve as 07-01 V23 (0.97315).
3. Achieve a similar number of false positives as 07-01 V23 (4).

6.12.4 *Measure of Success (08-01 - V28)*

The ESN model is expected to produce a similar response overall and a similar or better area under the ROC curve while maintaining or improving the classification rate and maintaining a similar number of False Positives.

6.12.5 *Data Description (08-01 - V28)*

Training Data; concatenation of:

- Partial 0813-1 LA (laboratory) – Subset from end including two mounts
- Partial 0716-2 RA – Central (riding) region reduced

- Partial 0912-1 RF – Central (riding) region reduced
- Partial 0912-2 RG – Central (riding) region reduced
- Partial 0913-3 RF – Central (riding) region reduced

Data from the riding phase was removed from each dataset in larger amounts compared with the initial 02-01 model. Only two sections of 1,000 samples of riding phase data were included from each dataset, 1,000 from immediately after the mount and 1,000 immediately preceding the dismount.

Testing Data; concatenation of:

- Full 0719-1 RB
- Full 0812-2 RA
- Full 0829-1 RC
- Full 0830-3 RD
- Full 0902-1 RE
- Full 0902-2 RC

The entire data file was always used for testing files.

6.12.6 *Static Parameter Description (08-01 - V28)*

The following static parameters were used during this iteration:

- Error Cost Function: (conditional mean square error class + conditional mean square error non-class)$/2$
- Included Sensors: Gyroscope and Accelerometer
- The Gyroscope and Accelerometer sensor readings are linearly transformed, real-values ranging between 0 and 1.
- The Gyroscope data was passed through a second order high-pass Butterworth filter with a frequency of 0.01Hz, designed to remove signal drift.
- The detrended Gyroscope data was then transformed with a rolling mean.
- No outliers were removed from either sensor, however the rolling mean applied to the Gyroscope data would tend to dampen the Gyroscope outliers.
- The three axis of Accelerometer data are combined into a single vector of values by summing the squares of each axis then taking the square root of the product.
- The single Accelerometer vector is then transformed into two vectors by using a Butterworth filter to separate the signal into upper and lower frequency bands.
- The Lukoševičius code automatically adds an additional input vector with fixed values of 1.0

6.12.7 *Run time Parameter Description (08-01 V28)*

The parameters from iteration 02-01 except for the number of neurons were used for this design iteration and are shown in table 6.46. A decision was made to increase the number of neurons to 1,000 as most of Lukoševičius's models used this number of neurons. The ESN model that was built for this set of design iterations was built, trained and tested on a Windows 7 based PC running an i7 processor and 8GB RAM.

Table 6.46: Run Time Parameters for ESN iteration 08-01 V28.

Description	**This Iteration**	Justification	02-01 Iteration
Input values	5+1 vectors of data	Input choice	3 vectors of data
Classes	Mount only	Class choice	Mount only
Number of neurons	1000	Lukoševičius (2012)	336
Input scaling	0.9252	From 02-01	0.9252
Neuron leak rate	0.1379	From 02-01	0.1379
Regularisation	0.0001	From 02-01	0.0001
Spectral Radius	0.9455	From 02-01	0.9455

6.12.8 *Testing Results (08-01 V28)*

Table 6.47: Processing Times for Concatenated Test Datasets iteration 08-01 V28.

Iteration Id	**08-01**	07-01 V23
User Time in seconds	441.00	31.47
System Time in seconds	00.08	01.11
Elapsed Time in seconds	441.19	32.64

Table 6.48: Classification Metrics for Concatenated Test Datasets iteration 08-01 V28.

Iteration Id	**08-01**	07-01 V23
ESN Test Error	0.3269	0.2373
RMSE	0.0378	0.0299
AUC	0.9970	0.9732

User processor time and elapsed processor time have both increased substantially. partly this can be attributed to the extra neurons in the model but also reflects that this code has not been optimised for R, where as the Schliebs code has been optimised. This also demonstrates that while the end effect of the code is the same or similar, the coding statements that are used in each case are quite different. producing similar results via two different sets of code provides some confidence that the underlying algorithm for implementing an ESN classifier is the same and the results produced are repeatable.

The error recorded from the test run is broadly similar between both iterations, especially considering that this latest iteration used more than twice as many neurons.

Table 6.49: Results for Concatenated Test Datasets iteration 08-01 V28.

Measures at 0.6 Cut-Off				
Description	**08-01**		07-01 V23	
Classification rate	0.714		0.571	
	Base Cl	**Mount Cl**	Base Cl	Mount Cl
Precision	0.9997	0.5065	0.9995	0.8462
Recall	0.9998	0.3451	0.9999	0.1947
Confusion Matrix at 0.6 Cut-Off				
	Predicted 08-01		Predicted 07-01 V23	
Actual	**Base Cl**	**Mount Cl**	Base Cl	Mount Cl
Base Cl	218022	38	217706	4
Mount Cl	74	39	91	22

Again, the RMSE of both iterations are broadly comparable and the RMSE for the current iteration is slightly higher, this is consistent with the response from 07-01 V27 where the larger number of neurons also resulted in an increase in the RMSE. The area under the ROC curve is similar between both iterations with the current area increasing to 0.997. This increase may well be attributable to either the larger number of neurons, the addition of the static values vector or both.

The precision of the current iteration is substantially less than that of 07-01 V23 at the 0.6 cut-off level but this is balanced by a corresponding large increase in recall at this cut-off level. This behaviour is consistent with a model that generates a more responsive output and is similar to what was seen with 07-01 V27. The confusion matrix at this same 0.6 cut-off level is consistent with the changes in both precision and recall, with the number of False Positives at 0.6 increasing to 38 but this is balanced against the increase in true Positives at this level and notably, the classification rate for this iteration has increased to 0.714. Figure 6.45 is very similar in shape to Figure 6.32 on page 211 for iteration 07-01 V23 and is an even closer fit to Figure 6.42 on page 219 for iteration 07-01 V27. The mounts occur where they are expected to be and even the false mounts occur where they are expected.

The individual mounts shown in Figure 6.46 illustrate that four of the six mounts that were expected to be classified were classified with a strong response that went well beyond the 0.6 cut-off, one mount only just made the 0.6 cut-off while the model response for the sixth mount was just below the 0.6 cut-off and would have been included if the cut-off was set at 0.5 instead.

The ROC curve (Figure' 6.47 a) demonstrates that this model is clearly better at distinguishing mounts at response levels below 0.2 and as good or better at response levels above that. This can also be seen in the shape of the figure that plots Precision against Recall (Figure 6.47 b) and Figure 6.47 c, that plots the number of True Positives against the number of False Positives at various cut-off levels.

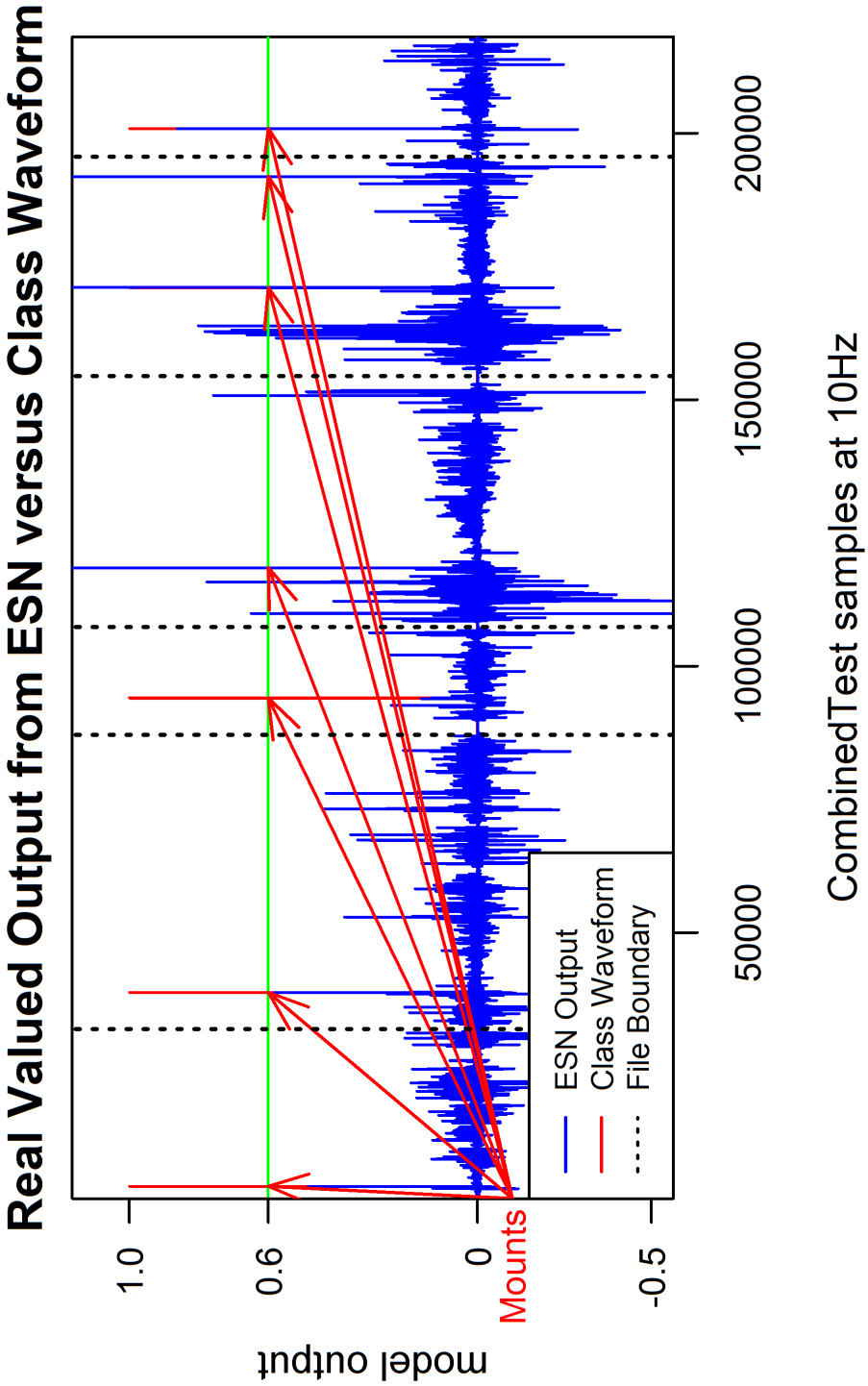
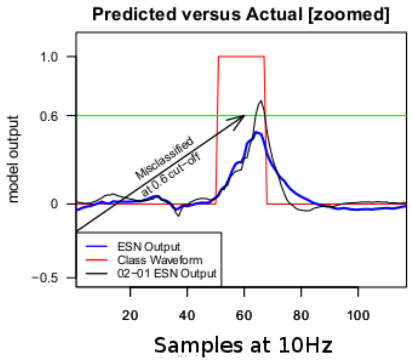
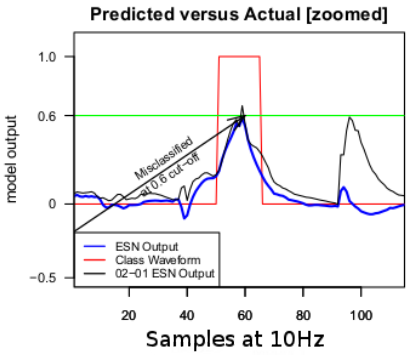


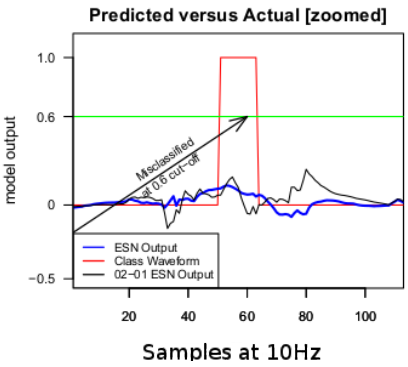
Figure 6.45: 08-01 V28 ESN Output for Concatenated Test Data



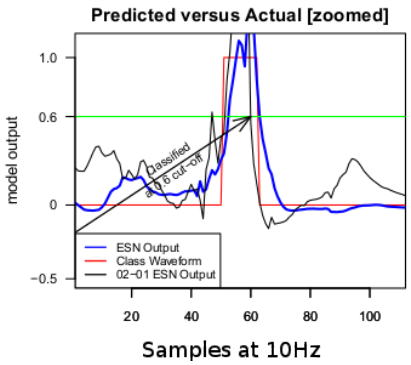
(a) 0719-1 RB #1



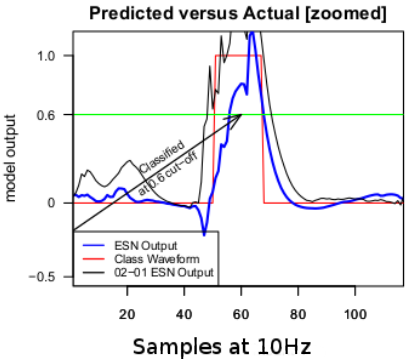
(b) 0812-2 RA #1



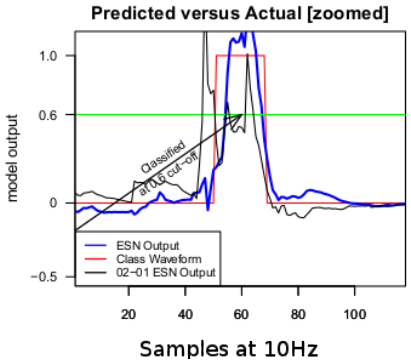
(c) 0829-2 RC #1



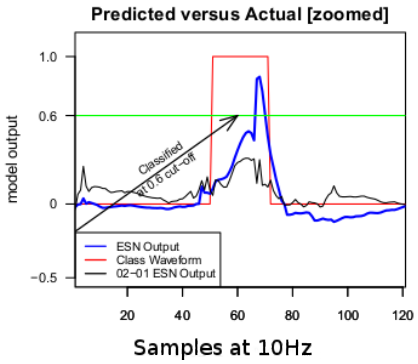
(d) 0830-3 RD #1



(e) 0902-1 RE #1

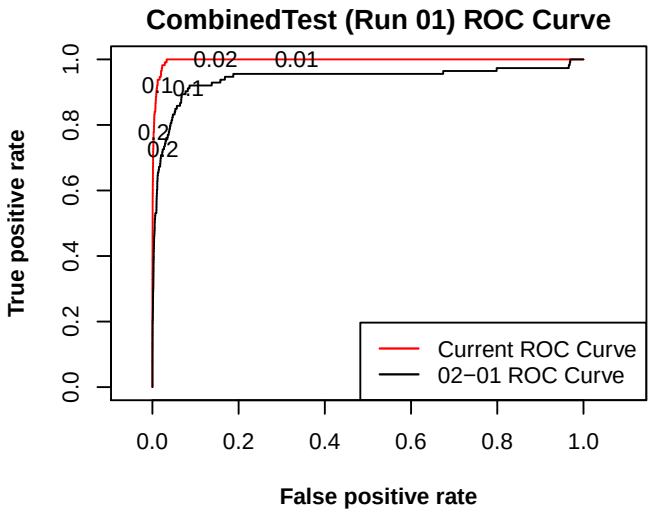


(f) 0902-1 RE #2

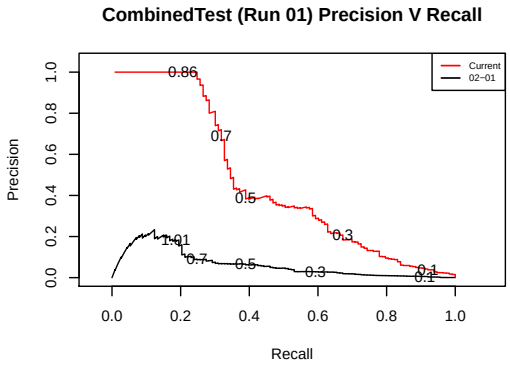


(g) 0902-2 RC #1

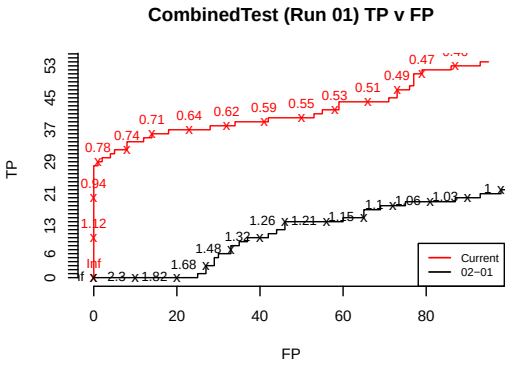
Figure 6.46: 08-01 V28 Mounts for Concatenated test data



(a) ROC Curve



(b) Precision V Recall



(c) TP V FP

Figure 6.47: 08-01 V28 Performance Plots for Concatenated Test Data

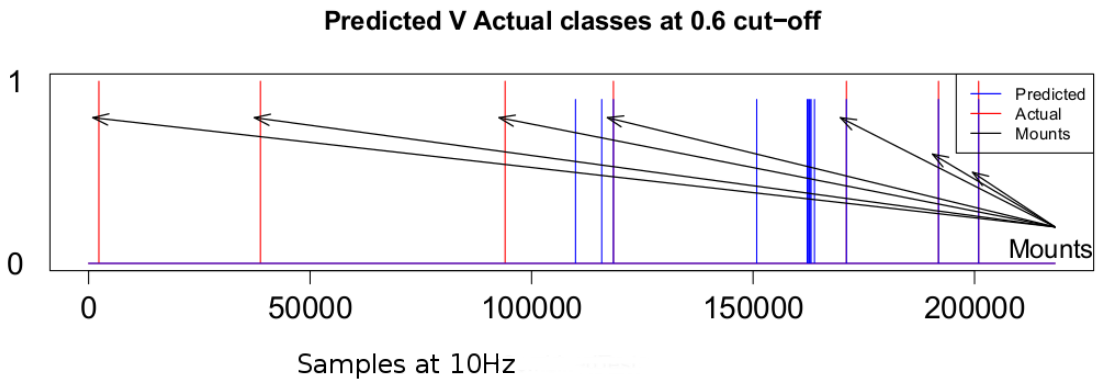


Figure 6.48: 08-01 V28 Classes for Concatenated Test Data

6.12.9 *Results versus Goals (08-01) V28*

This iteration is considered to be similar enough, given the differences in model size and the additional input vector, to have met its design goals.

6.12.10 *Discussion (08-01) V28*

Although the goal of this design iteration was not to improve the software artefact, it did result in a larger area under the ROC curve. This reinforces that while this research has produced an artefact that is good enough to meet the overall research goals, there is still a lot more knowledge to be gained about using ESN recurrent neural networks to classify punctual activities.

6.13 SUMMARY OF ITERATIONS

6.13.1 *01-01 – Run ESN from phase two with real-world data*

Section 6.4 on page 147 describes iteration 01–01, the first iteration in this series and in this first iteration using real–world, unscripted activity data the author decided to test the best performing ESN model from the phase two work against the real–world data in a somewhat simplistic attempt to see if the model developed from the scripted, laboratory based data had validity when used with real–world data. Curiosity rather than any real expectation of successful classification carry over drove the interest for this iteration and so it was not disappointing that the results showed that the model developed for scripted data did not do a particularly good job of classifying the data from the unscripted activities and in particular the model generated a huge number of False Positive classification errors. However the model successfully classified both the Mounts and Dismounts that were in the test data and this was pleasing. While these successful True Positive classifications could well have been somewhat of a fluke (given the high number of False Positives) these correct classifications are encouraging and hinted that the underlying technique of using an ESN classifier might well successfully transfer from scripted to unscripted activities.

Iteration 01–01 was a transitional situation and so the data used for testing this model has the same basic characteristics as the data from the scripted activities that were used to train this model. As a result the test data includes all nine sensor outputs and sets out to classify both Stirrup Mounts and Dismounts. In addition the cost function used during both the training and the testing of the model uses a relatively common least mean square error function.

6.13.2 *02-01 – Build new optimised ESN model using real-world data*

Section 6.5 on page 149 describes iteration 02–01. The work described in this section was published in Hunt and Parry (2015). This was the first ESN model built specifically for the data captured from the unscripted, real–world riding activities and was

used as the base model against which to compare the following design iterations. This model incorporated much of the learning that had been gained from the phase two activities. Following the simplification design strategy, the process of producing, tuning and testing the ESN model was simplified and standardised. A Debian based, multi–core, virtual server, set up within the Amazon Elastic Cloud Compute environment, running R and RStudio was used for this and subsequent design iterations. This platform then became the standardised (and easily cloned) platform for running the PSO parameter search during the tuning cycle for each subsequent model.

During this third phase of the design cycle the data was always split into three separate groups, training, tuning and testing. The training data was used to train the ESN model. The tuning data was used during the parameter optimisation process to test against the trained model and finally the testing data was held back to the last step so that the model can be tested with data that has not previously been seen. This data is listed in Table 6.1 on page 147.

One of the important things that was learned during the phase two activities was the difficulties associated with classification of rare classes. This showed up in two key areas. Firstly, without some technique of introducing a bias into the model training process there was a tendency for the ESN model to be optimised for the much more common base class. During this third phase two techniques are used to try to counter this tendency towards optimising for the base class. The first technique was to take out (under sample) some of the base class data from the training and tuning datasets (but never from the testing dataset). This has an added bonus of speeding up the model development and parameter tuning process as there was less data to deal with during these parts of the process. The second technique that was used to bias the model towards the much rarer non–base class data was to use a conditional mean square error as the cost function to be minimised during tuning rather than a straightforward mean square error cost function.

In line with the simplification strategy, a decision was made to only classify Mounts rather than both Mounts and Dismounts. One of the advantages of using an ESN model for classification is that the same reservoir can be used to classify a number of different classes in parallel but at this stage of development of the optimisation process for the ESN model parameters there is no simple technique of balancing potentially opposing parameter settings for differing classes. Of course, if classes have similar parameter requirements then this is not an issue. Every research project must ultimately decide what needs to be left out just as importantly as what should be included and in this case for this project a decision was made to leave out the classification of multiple classes at this time. This is likely to be one of many future directions that are looked at for post-doctoral work. One immediate benefit of this decision was that the model development and tuning process was simplified and sped up. The modified, shorter version of the Mount definition was used during this phase of the work, please see Section 5.6 on page 127 for that redefinition.

The last major simplification that was applied to this phase of the work was to start by using only the Gyroscope data as this data was the least affected by orientation changes and noise. The intention was to start with only the Gyroscope data and to try to get as much value out of the Gyroscope data before adding any other sensor data. If

it becomes possible to consistently classify the Mount activity with only Gyroscope data then the development process will stop there but if consistent classification was not possible using only Gyroscope data then data from the other sensors will be added to see if that helps. This is consistent with the iterative development cycle of Design Science.

6.13.3 *03-01 – Compare ESN model using offset input data against 02-01*

Section 6.6 on page 160 describes iteration 03–01. The primary purpose for this design iteration was to look at a suggestion in Jaeger (2005, p. 44) to beware of symmetric input. According to Jaeger, when the input values are roughly symmetrical around zero then the ESN using a sigmoid actuator may not learn effectively. The Gyroscope signal is very much symmetrical around zero. The input signal used with this iteration has a constant offset added to it so that it is no longer symmetrical around zero and is instead offset from zero.

6.13.4 *04-01 – Compare ensemble ESN model against 02-01*

Section 6.7 on page 169 describes iteration 04–01. The design goal for this iteration was to test a simple ensemble using both the gyroscope data as input plus the output from a much smaller, 60 neuron, ESN. The idea for this iteration came from comments by (Lukoševičius, 2012, p. 669) on the possibility of splitting a reservoir into different populations with differing parameters as a way of dealing with multiple time scales within the same input stream. A decision was made to implement this idea by including the output from a much smaller model, rather than reprogramming the ESN library code to enable the splitting of the main reservoir into two or more populations with differing parameters. This alternate approach to including differing time scales is consistent with (Lukoševičius, 2012, p. 675) and other authors such as Jaeger and Haas (2004) who report that ensembles of ESN can sometimes dramatically improve performance over a single ESN.

The parameter optimisation process for this iteration produced a parameter set that was noticeable different from the prior models and the optimised ESN produces a much flatter output overall that reduces the number of false positives but also reduces the number of true positives that it classifies. Also of interest was that this ESN correctly classifies one mount during the final test that had not been classified during the two prior iterations.

The experience gained during the tool development phase and the initial work done during this phase lead to an initial conclusion that the parameter optimisation process for 04–01 may have got itself into a local minima and so this iteration was repeated with a different set of parameters to see what difference this will make to the ability of the ESN to correctly classify the test data. This was reported as iteration 04–02.

6.13.5 *04-02 – Compare an alternate ensemble ESN model against 02-01 & 04-01*

Section 6.8 on page 180 describes iteration 04–02. This iteration was similar to iteration 04–01 except that it builds and tests the ESN model using parameters from a similar iteration. The parameter set from a different, but somewhat similar earlier model was chosen rather than re-running the parameter optimisation process with a longer termination generation because the parameters from the alternate model are closer to the expected parameter values and also because running a longer optimisation process would take much longer and time to complete the research was quickly running out. The classification ability of the ESN model is much more in line with iteration 02–01 and 03–01 when set up and tested with this alternate parameter set and it is not particularly remarkable.

6.13.6 *05-01 – Compares an ESN model trained with under-sampled input data against prior models*

Section 6.9 on page 190 describes a set of design iterations that was developed using under-sampled training data. Under-sampling seems to improve classification and as a result under-sampling was used with the design iterations that followed.

6.13.7 *06-01 – Compares an ESN model with additional Gyroscope features against prior models*

Section 6.10 on page 198 describes a set of design iterations that will be developed using different features generated from the the Gyroscope data using a Butterworth filter. These features seem to improve classification and are used with the design iterations that followed.

6.13.8 *07-01 – Compares an ESN model with Accelerometer features against prior models*

Section 6.11 on page 206 describes a set of iterations that add Acceleration data to the Gyroscope data to see if this improves classification. As was noted during the second, tool development phase, the raw Accelerometer data as three separate axis are somewhat inconsistent and so for this iteration set the author decided to use the "net acceleration power" instead of the individual axis. The net acceleration power is calculated by summing the squares of each axis then taking the square root of the product. The resultant number is then scaled to be between 0 and 1.

The addition of the additional information in the form of the net acceleration power substantially improved the classification of the Stirrup Mount activity using the ESN model. The improvement in classification was such that this model was considered good enough to demonstrate that the overall classification goal for this research had been achieved.

6.13.9 *08-01 – Compares an ESN model using alternate R code against 07-01*

Section 6.12 on page 224 describes design iteration 08-01. This iteration was essentially an attempt to demonstrate the robustness of ESN classification of punctual activity by using an alternate R code library written by Mantas Lukoševičius, an ex–student of Jaeger's. Lukoševičius had posted some public code on the internet in various languages with a simple implementation of an ESN. A copy of the "R" code, once an error within it was fixed, was run against the data used in iteration 07-01. The results are very similar to those obtained from the original code.

Chapter 7

DISCUSSION

This chapter discusses the results of the iterative development done during the three-phased development of the Punctual classifier of the real-world equestrian sport activity and the knowledge gained during that development. It also discusses the implications and limits of that work.

7.1 INTRODUCTION

This research set out to design and construct an ACS to classify punctual activities of interest within Equestrian Sport and in particular, to construct a working system using real or realistic data so that a RC based punctual ACS could potentially be used within a wearable coaching system for equestrian sports-people. This goal was achieved and described in Chapter 6 where a RC based classifier that is capable of classifying non-windowed, real world, punctual activities with possibly divergent temporal activity frames across multiple subjects in differing geographic locations and on different session dates was demonstrated. Within Chapters 5 and 6 the author discussed elements relevant to the particular phase or design cycle that was being described. The iterative development cycle that forms part of the Design Science methodology used within this research has enabled the author's understanding and knowledge of activity classification to grow as the research progressed. Reviewers will note that the sophistication of the work and the analysis has developed in tandem through the development chapters of this document. It is also worth reinforcing that this work did not set out to claim that RC techniques or ESN in particular are better than alternate techniques of classifying punctual activities but rather that RC techniques and ESN are capable of being used for this purpose. It is possible that RC techniques may well be superior in some areas but proving this is something for future work. This chapter discusses the wider issues that came up while this research was being conducted.

7.2 RESEARCH QUESTION REVISITED

The work described in section 6.11 demonstrates that it is possible to construct a RC based classifier that is capable of classifying the activity of interest when presented with real, unscripted, un-windowed data captured from a wrist mounted inertial sensor.

In additionsection 5.2 demonstrates that it is possible to construct a RC based classifier that is capable of classifying complex, un-windowed spatio-temporal data as a proof-of-concept; and section 5.4 demonstrates that it is possible to construct a RC based classifier that is capable of classifying scripted, punctual activities using realistic, un-windowed inertial data with reasonably consistent temporal frames.

Finally, section 6.11 demonstrates that it is possible to construct a RC based classifier that is capable of classifying unscripted, punctual activities using real, un-windowed inertial data with divergent temporal frames.

7.3 CONTRIBUTION

This work was successful at demonstrating a classifier that was able to reliably classify stirrup mounts from real-life horse riding sessions, a complex task given the rarity of mount data within a typical real-world riding session and the differences in techniques across different riders. In constructing and demonstrating this classifier a number of contributions have been made to knowledge. Firstly, to the best of the author's knowledge, no other researcher has demonstrated a classifier capable of reliably classifying this particular activity and so a contribution has been made to the general area of equestrian sport.

Secondly, this research has demonstrated the ability of RC based neural networks to classify human punctual activity based on spatio-temporal inertial sensor data. The author believes that this is the first time that RC systems have been used to classify human activities in this manner. Of course, RC systems have been used successfully as classifiers in other areas but not as inertial data based human activity classifiers. This research has not attempted to measure a RC based human activity classifier against more traditional human activity classifiers and so no comparison may be made between this RC based classifier and another classification technique in this area however, it is useful, none-the-less, to demonstrate to other researchers that RC techniques are a possible alternative and part of a wider activity classification tool-kit.

This work has highlighted the difference between durative and punctual activities and has pointed out the sparsity of research in the area of punctual activities classification versus durative activity classification. This is not the first work to have highlighted these differences but the prior work in this area has been both somewhat light and not altogether clear and so this work has extended those earlier works and added clarity in this area. This added clarity combined with the sparsity of work on punctual activity classifiers may well encourage other researchers to extend this work on punctual activity classification and to move into other related areas.

This work was successful using relatively simple features extracted from the input data whereas many other researchers, particularly those with an interest in durative activities typically use much more complex features calculated from their input data. As a result, and combined with the intrinsic "memory" that is a feature of RC based classifiers, other researchers may be tempted to re-explore simpler features in their own research.

Looking somewhat wider into the related area of data mining, the ability of the ESN to successfully classify rare activities may mean that rare data mining techniques based on ESN may have some merit for researchers in those areas.

7.4 SUMMARY AND GENERALISATION OF RESULTS

Throughout the iterative development cycles that are a fundamental part of the Design Science Methodology each individual iteration described the goals for that iteration, described the set-up, running and results and then discussed that particular iteration and came to a conclusion. While the insights sit best within the environment that created them it is also useful to summarise the main discussion points here so that the reader has access to them in a concise, combined situation. This also aids in generalising some of the discussion points. Some of these discussion points that are considered to be key or important will also be discussed in more detail in following sections.

The first area to summarise here is the chosen methodology and its effects on the work. In this discussion Design Science will be contrasted with the Experimental methodology. This is not meant to be an exhaustive discussion or a complete contrast between the two methodologies and many other researchers have done that before and have done it much better than the author can. Some of these include Wynekoop and Conger (1990), Jarvinen (2000), Kjeldskov and Graham (2003), Hevner et al. (2004), Winter (2008) and Vaishnavi and Kuechler (2015) and the reader is pointed to these works if they want a more in-depth discussion.

Without going into a lot of detail, Design Science can be said to be a methodology that covers a wide area of interest at a relatively low depth and which is designed to produce something that has utility by working, rather than setting out to find the truth about something or to find the best way of doing something. In a sense it could be compared to the PSO search algorithm that was used during the mid part of this work. Different aspects of, in this case development, are tried and the results are noted. Areas that seem to offer positive results are kept while areas that seem to offer negative results are dropped and then a path is sort through the search space that links the positive areas.

The end artefact produced via Design Science has utility because it works and anecdotal knowledge is acquired about promising areas for future, more in depth, studies. However, the down side of this methodology is that it is not possible to say that the artefact is the best possible artefact overall or even that the areas that provided positive contributions to the artefacts utility are configured in the most optimal way nor even that all areas that can contribute positively have been identified. This also makes it dangerous to attempt to generalise the results too much. Statements such as "*We found that reducing the drift in the Gyroscope data seemed to have beneficial results for classification when we looked at these X datasets while attempting to classify Z activity using the W classification engine but we are unable to either quantify the benefits nor can we assume that this will always be the case*" are possible when using Design Science but more imperative statements such as "*We found that reducing the drift in the Gyroscope data was beneficial for classification and future researchers*

should ensure that this is done before attempting classification based on Gyroscope data" are not possible because there is no provable foundation for them. As a result, the following discussion highlights areas that are worth looking at when attempting to do something similar but there is no *recipe* to follow that comes out of this work.

7.4.1 *Synthetic Activity Data and Liquid State Machine*

Section 5.2 on pages 95 through 101 describes the attempt to demonstrate that a RC technique, LSM, could be used to classify synthetic inertial activity data and it follows a similar process to that described in Maass et al. (2002). Synthetic inertial data was generated, a known random noise profile was added to the data and two LSM classifiers were trained on half of that data using differing spike encoding techniques. Both models had a similar training profile but when presented with previously unseen data only the model that used Population encoding for the input spike train had an acceptable classification result (93.5% accuracy), versus 24.7% accuracy for the BSA encoded model. The discussion relating to this design iteration is shown in Section 5.2.10 on page 101.

While the Population encoded input spike train data was demonstrated to have reasonable classification accuracy and the BSA encoded input spike train did not demonstrate acceptable accuracy the BSA encoding technique should not be written off as being unacceptable in other circumstances as it is possible that a differently configured reservoir may produce more acceptable classification. On the other hand, the Population encoded data did produce promising results in this particular situation and considering the short training period this was even more encouraging. Most of the classification errors from the Population encoded data occurred at the start of the activity and it is suggested that this might be as a result of the classifier needing to recognise some of the characteristic data associated with the activity and accumulating it before generating enough "energy" to respond with a recognition spike, however there may be other possible reason for this.

7.4.2 *Laboratory Activity Data and Liquid State Machine*

Section 5.3 on pages 102 through 112 describes the attempt to demonstrate that a RC technique, LSM, could be used to classify realistic inertial activity data that was collected in a laboratory situation with a single participant following a set script of activities. Following on from the prior iteration, this iteration uses only Population encoded input data and a one second buffer (10 samples at 10Hz) before and after the activity of interest to try to pre-load the classifier so that has accumulated enough relevant signal within the reservoir to start classifying the activity at its actual boundary. All nine vectors, three by three sensors (Accelerometer, Gyroscope and Magnetometer), of data are used as input. A drift is noted within the Gyroscope and Magnetometer data but is not corrected. The input data is cleaned and normalised prior to encoding into spike trains. The LSM model produces satisfactory results with 85.1% of samples correctly classified. A sample is considered correctly classified if the model output is above 0.6 for the classes of interest. False Negative errors

tend to occur at the start of each activity and False Positive errors occur towards the end of the input data stream where the LSM model encounters activity data it did not see during training.

The discussion for this iteration is in Section 5.3.12 and starts on page 111. The classification accuracy for this iteration met the design goals and is considered satisfactory. However, it should be noted that the data is collected within a laboratory situation with the participant following a set activity script and so the script itself introduces signal artefacts into the input data and as it is not possible to tell what data is influencing the LSM model when it successfully classifies an activity it is possible that it is the script artefacts that are being used for classification instead of the underlying Mount/Dismount behaviour. In addition, the misclassification of the False Positive towards the end of the input data highlights that only data directly associated with Mounting and Dismounting has been captured and tested against and so it is impossible to predict how well this classifier will react when it encounters data from wider equestrian activities. There is also no way of predicting how well this classifier will react to data from a different participant as this has not been tested. Lastly, LSM models have a large number of meta-parameters that need to be set so that they function as desired and setting these variables to a useful value is a complex and long winded process with no agreed heuristics for simplifying this process. This means that using a LSM model is difficult for most researchers who are not experts in this area. This complexity triggered a re-examination of the tools that would be used within the rest of this work.

7.4.3 *Exploration Phase*

Section 5.4 on page 112 through 130 describes the work done to build a usable researcher's workbench for Punctual activity classification. A PSO (Back & Schwefel, 1993; Bendtsen., 2012) search was swapped for a grid search to help speed up the RC model meta-parameter search process. A change was made from Python to R and RStudio to allow for reproducible research (Stodden, Leisch, & Peng, 2014) and an integrated development environment with a number of complementary modules including easy parallelisation, matrix operations, the PSO search process and the ready availability of other classification engines (Lantz, 2013) including SVM and HMM and Random Forest. Parallelisation was implemented on a multi-core virtual machine that was available on demand utilising Amazon Elastic Cloud Compute. Lastly, the classifier engine was switched from the more complex to implement LSM to the less complex ESN (Verstraeten et al., 2007).

In addition the datasets were split into three subsets rather than two as per findings from Flexer (1996) and Henery (1994). One subset was used for training during the meta-parameter tuning phase and the final test phase, another subset was used for testing during the meta-parameter tuning phase and the final subset was set aside and only used during the final test phase. This resulted in developing a very easy to use researcher's workbench that was capable of being run locally on MS Windows, MacIntosh and Linux or remotely on the same three operating systems. The implementation of ESN using the matrix operations inherent with R, the parallelisation

of this code and the implementation of the PSO search cut model development time down from a week or more to typically less than six hours. The use of R's reproducible research features, its ease of integration with latex and HTML and its strong graphing features helped make documenting the development simpler and GIT Hub made managing changes and versions simpler.

This work set up the research process well for the next phase and was culminated with a reproduction of the prior work done in Section 5.3 using the new tools. The results of this work are described in Section 5.4.15 on page 117 and the comments are in Section 5.4.16 on page 119. The new tools were easier and simpler to use and took considerably less time to run, end to end. Good classification rates were achieved but given that the data was scripted and recorded in a laboratory environment it is difficult to assess if the classifier is using signal artefacts from the activities of interest or from other facets of the script and so there is little confidence that this classifier would work equally well with real world, unscripted data. This is supported by similar comments from Bao and Intille (2004) and Ravi et al. (2005).

Following the work done to develop the researcher's tool bench the activities of interest were redefined to make the real life versions more consistent and following this the new activity definitions were tested against the prior ones for consistency. This process is described in Section 5.8 on pages 130 through 141. The results of this iteration show that the new activity definitions achieved comparable results to the prior definitions on the scripted, laboratory data. The discussion for this iteration is in Section 5.8.11 starting on page 140.

Within this part of the work it was noted that training an ESN model with only the three vectors of Gyroscope data gave almost as good results as using all nine vectors. While the Magnetometer data showed some informational value within the laboratory data which was all collected in the same geographic position starting and ending with the same compass bearing it is argued (see Section 4.6.5, page 89) that without transforming the Magnetometer data in some way that it would not be usable within the real world data. ESN models with more than 1,000 neurons tend to be more easily over-trained and when using ESN with only three vectors of input, models over 500 neurons are easily over-trained. This means that when searching for optimal meta-parameters most searches should not bother trying more than 500 neurons for three vector input data models. In general adding small numbers of neurons adds to classification ability linearly but increases computing resource usage logarithmically. Some ESN classification models with only 50 to 60 neurons produced surprisingly useful results and as a result it is possible to consider using these smaller and much less resource intensive models in some situations, perhaps as an ensemble of classifiers.

In some situations when using input data that is relatively symmetrical around a zero mean such as the Gyroscope data then adding an offset to the data as suggested by Jaeger (2005) can have beneficial results for classification. When dealing with highly imbalanced classes such as those encountered for mounting in the real world then the use of mean square error as a cost minimisation function can produce results that are highly biased towards the majority (base) class and a more suitable cost function can produce more balanced results for both the base class and the activity

of interest. Tuning and testing a ESN model that only needs to consider one class other than the base class substantially reduces the required computing resources and produces results that are easier to compare, model on model especially when using the area under the ROC curve for comparisons. The preferred measure of one model against another within this is is the Area Under the ROC Curve. This measure has been chosen as it is relatively immune to class skew and based on advice from Bradley (1997) and Tang, Zhang, Chawla, and Krasser (2009) considering the wide class imbalance between the base class and the class for the activity of interest.

As described in Section 6.3.2 on page 146 the activities of interest, Mounts and Dismounts, are trigger events and so within this work True Positive and False Positive classification results are given prominence over False Negative results. Provided the RC model output is strong enough to register at least one point within the range of the activity of interest then the trigger will be recognised and so when evaluating the model response no consideration is given to False Negatives.

7.4.4 *Laboratory Model with Real World Data*

This iteration is described in Section 6.4 on pages 147 through 148 and the results are discussed in Section 6.4.2 on page 148. This iteration tests the ESN model that was trained on the laboratory data is Section 5.8 against the real world riding data. The goal was to see if the laboratory data was similar enough to the real world data that successful classification could occur. There was no real expectation that this model would successfully classify the real world data as per Bao and Intille (2004); Foerster et al. (1999); Ravi et al. (2005) but it was decided to at least test this assumption.

As expected, the laboratory trained ESN classifier was not successful in reliably classifying the real world Mount and Dismount. The classifier produced a very large number of False Positives however the classifier did manage to to classify both the Mount and Dismount at a 0.6 cut-off level. The ratio of False Positives is so high that this classifier is unusable on real world data and confirms the earlier assumptions that a classifier trained on a small set of scripted activities quickly breaks down when it encounters a previously unseen wider variety of real world situations.

7.4.5 *Echo State Network Model with Real World Data*

This iteration is described in Section 6.5 on pages 149 through 159 and the results are discussed in Section 6.5.12 on page 159. This iteration classifies Mounts only based on a ESN model that is trained on three vectors of Gyroscope data from three files containing real world riding activities, has the meta-parameters of the ESN classifier tuned using a separate three files containing real world riding activities and then the ESN optimised model is tested on a further, separate five files containing real world riding activities. This ESN model and its results then form the base for comparing future design iterations against.

This ESN classifier achieved an area under the ROC curve of 94.18%, a very satisfactory result. Five of the seven Mounts were correctly classified at a 0.6 cut-off point, however at this same cut-off point there were also 415 False Positives. Given the rar-

ity of the Mount data (less than 0.06%) and the small number of training datasets this is a satisfactory result and provides confidence that a ESN classifier engine is capable of reliably classifying this rare, punctual activity.

7.4.6 *Echo State Network using Offset Data*

This iteration is described in Section 6.6 on pages 160 through 168 and the results are discussed in Section 6.6.12 on page 168. The potential usefulness of a data offset for input data that is symmetrical around a zero mean was noted during the review of knowledge learned during the process of developing the Researchers Workbench and is supported by the work of Jaeger (2005). This iteration looks to see if adding an offset to the Gyroscope input data will improve classification.

The results of this design iteration did not demonstrate any improvement in classification and, in fact, when measured by the area under the ROC curve (86.97% for this iteration Vs 94.18% for the initial, base, real-world data iteration) the performance of this classifier, when using optimally tuned meta-parameters, dropped considerably. This is slightly at odds with the advice of Jaeger (2005), however no attempt was made to manipulate the amount of offset and so this configuration facet of ESNs is only partially tested and as a result this work offers no advice on this one way or the other. However, within this work, no offsets were added to input data in future design iterations except for the very last design iteration where code from one of Jaeger's students was used (Lukoševičius, 2012) and instead of offsetting the input data another vector with a constant high value is added. This alternative design innovation seemed to have had positive effects. Please see the comments on the last design iteration for more on this.

7.4.7 *Ensemble Echo State Network - iterations 1 & 1A*

This iteration is described in Section 6.7 and Section 6.8 on pages 169 through 188 and the results are discussed in Section 6.7.11 on page 178 and Section 6.8.12 on page 188. A number of authors including Chawla (2005), Z. Zhou and Liu (2006), Mollineda et al. (2007) and X. Liu et al. (2009) have suggested using ensembles of classifiers, especially with imbalanced classes and so this design iteration acknowledges the learning relating to the usefulness of small ESN classifier discussed in Section 6.1 and looks at a simple ensemble that uses the output of a smaller ESN classifier as additional input into the main ESN classifier.

The meta-parameter tuning process for this classifier produced a somewhat unusual set of run time parameters compared with prior classifiers. In particular the leaking rate remained high, tending to indicate that this ESN classifier was more influenced by the current input rather than the earlier input, indicating that this classifier was less influenced by the temporal order of the data and was more akin to a simple feed-forward neural network. The output from this ESN model tended to move above zero and demonstrated less variance than the output from prior models. This resulted in considerably less error for the base class with an expected zero output and slightly less False Positives but this tendency of less variance from zero meant that there were

also less True Positives for the Mount class. Overall there was was an increase in the area under the ROC curve.

This model presents a dilemma, on one hand, using the area under the ROC measure the addition of the additional input vector has improved performance but the classification rate for the (rare) mount class has fallen and so less mounts are being classified. In many ways this represents the expected behaviour of a standard classifier when presented with highly imbalanced class data as this data is. The tendency is to simply classify everything as the base class as this reduces the majority of the error. However, prior versions of this ESN classifier have performed "better" in that while they may have more error in the base class they were better as distinguishing the class of interest. In addition, as a result of the tuned, run time parameters being so different from the earlier ESN classifiers it is difficult to tell if the changed response comes as a result of the run time parameter changes or the addition of additional information from the add on input vector.

Recognising that this dilemma where a major change in run time parameters at the same time as other changes are being made would make forming conclusions about the cause of any beneficial change difficult, a decision was made to eliminate changes in run time parameters by discontinuing the meta-parameter tuning process. This decision was supported by the observation that until this model, the tuning process had tended to produce very similar run time parameters anyway and this coincides with some of the observations of Lukoševičius (2012) where he recommends reasonably generic run time parameters for most ESN models and a suggestion that parameters perhaps only needed to be tuned for differing temporal scales. That is, it may be advantageous to tune for longer activities or shorter activities but when activities have a similar temporal scale then the parameters that suit one activity most likely will also suit the other, similarly scaled activity. From a design iteration point of view this has two additional benefits. Firstly it is no longer necessary to run and report on the meta-parameter tuning process and so some effort is saved and, if tuning is no longer required then the data which has previously been set aside for testing within the tuning process can, instead, be used to enhance the ESN training process that has been critically short of training data. With this in mind and to try to separate the changes from adding the ensemble classifier input from the parameter changes, another iteration was run that used run-time parameters much closer to what had been seen in earlier iterations.

The results from the second, alternate iteration which used run time parameters within the expected range showed that the area under the roc curve decreased, both when compared with the prior iteration and also when compared with the base real world data iteration. Once again, the number of False Positives dropped, they dropped slightly compared with the prior iteration but substantially compared with the initial, base comparison iteration. The number of True Positives remained the same as the prior iteration but are less than the initial, base iteration. Again the ESN output variance has been reduced but this has been to the benefit of the base class rather than to the Mount class. The overall result is a less desirable output from the ESN classifier, despite the increase in Mount training data.

In this instance, a decision was made not to use an ensemble classifier of this form within future design iterations within this work as the overall result was a reduction in the area under the ROC curve. However, an ensemble classifier in another form, perhaps as reported in "*A Clockwork RNN*", Koutnik, Greff, Gomez, and Schmidhuber (2014) may be tried as part of future research as this form of classifier holds the prospect of parallel classification of activities with multiple, different temporal scales and this is attractive because such a classifier could be much more efficient with processing resources in a wearable environment due to its parallel classification abilities.

7.4.8 *Increased Under Sampling with Echo State Networks*

This set of iterations is described in Section 6.9 on pages 190 through 197 and the results are discussed in Section 6.9.11 on page 197. Under sampling is an accepted technique for improving classification rates when dealing with rare classes and has been recommended by researchers such as Drummond et al. (2003), Weiss (2004) and Chawla (2005). Each design iteration within this research since (and including) iteration 02-01 has used under-sampling to reduce the data within the central, riding region of the recorded activities. In this series of three iterations the amount of under-sampling is progressively increased by including less of the central, riding region from each dataset. This set of design iterations does not have their meta-parameters tuned individual, instead they all use the same run time parameters as used within Section 6.5 on page 153 to make it easier to separate out the effects of the under sampling from possible effects from run time parameter changes. As no meta-parameter tuning is required, the testing files from the tuning process are included in the training process for the test classifiers, thereby providing slightly more training examples of the Mount activity.

All three iterations produced an area under the ROC curve that is less than the initial, base real world iteration with V13, with the largest amount of under sampling providing the best of the three iterations. The output from all three ESN classifiers was subdued (close to zero). On the surface this is a worse classification result, however, while the ESN output was subdued, it was still discriminant in that the output peaks when Mounts were occurring are lower but as the base class signal was also much closer to zero then it was still possible to distinguish some Mounts, albeit at a lower output level. In addition there was a substantial drop in False Positives at a 0.6 cut off level. Once again, this produces a bit of a dilemma, on the one hand the main measure, the area under the ROC curve has performed worse but on the other hand the level of False Positives has also dropped significantly from 415 in the initial, base, real world classifier to 104 in the V13 classifier. A decision was made to retain under sampling for its possible beneficial effects of reducing False positives while looking for additional ways of increasing the ESN output response when encountering the Mount class.

7.4.9 *Filtered Data with Echo State Networks*

This set of iterations is described in Section 6.10 on pages 198 through 205 and the results are discussed in Section 6.10.11 on page 205. As noted in Section 4.6.4 and in Figure 5.16 on page 131, the Gyroscope data exhibits a drift over time and this is a characteristic of some Gyroscope sensors that is present within the data used in this work. Also within Section 4.6.4 where the data capture sensor is described it was noted that there is considerable noise within the sensor signals that comes from a variety of sources. Given that this work tracks the signal amplitude over time as the only input vectors then any drift present within the signal will make the class learning process more difficult as levels of amplitude will change over time as a result of the drift rather than as a result of the underlying activities. In addition and as noted by Lukoševičius (2012) while ESN classification engines are somewhat immune to random noise at lower levels they can be adversely affected by larger quantities of noise especially non-random noise. The sensor description noted that some of the noise was related to radio interference and is not random in nature. As a result, this series of iterations is designed to look at what happens when the drift and the noise are removed or reduced. This is not a comprehensive set of experiments, instead these iterations are designed to give a feel for the usefulness of reducing the very low frequency drift and the high frequency noise.

It was decided to use a Butterworth filter to reduce the low frequency drift based on recommendations from Mathie et al. (2004), K. Y. Chen and Bassett (2005), Godfrey et al. (2008), Lau and Tong (2008), Avci et al. (2010), Mannini and Sabatini (2010), Plötz (2010), Godfrey et al. (2011), Anguita et al. (2013) and D. Liu et al. (2014). In deciding where to set the lower bound level for the Butterworth high pass filter then the advice from Mathie et al. (2004) was that the lower bounds for most human activity is 0.3Hz was considered along with C. Yang and Hsu (2010) who suggest that 0.25Hz is a possible lower bound. In addition, Godfrey et al. (2008) suggested a lower bound of 0.6Hz but as their work was solely with human gait it is not so relevant. K. Y. Chen and Bassett (2005) was the most specific and their recommendation was to set the filter lower bounds at 0.1Hz to (specifically) eliminate Gyroscope drift and so this is what was done. In addition, D. Liu et al. (2014) and Mannini and Sabatini (2010) both recommend a second order Butterworth filter and so that advice was taken. During the exploratory phase described in Section 5.4 the author had successfully used a rolling mean to remove noise and so the decision was made to also use that technique here rather than adding an additional low pass Butterworth filter to to make a combined band pass Butterworth filter. In part, this decision was also took account of two of the design strategies described in Section 4.4.4, on page 74, "*favour simplicity*" and "*be mindful of possible implementation issues*".

Three design iterations were created and run with and without the rolling mean. The best iteration, V15, was the iteration that used both the Butterworth filter and the rolling mean and it achieved the best results with an area under the ROC curve of 96.94% and only resulted in 36 False Positives at the 0.6 cut-off level. This is much better than the results from the initial, base, real world classifier and so filtering to remove drift and a means of reducing noise were carried through to future work.

7.4.10 *Gyroscope and Accelerometer data with Echo State Networks*

This set of iterations is described in Section 6.11 on pages 206 through 222 and the results are discussed in Section 6.11.12 on page 221. This set of iterations was designed to build a series of new ESN classification models based around the idea of adding additional input vectors based on derivatives calculated from the accelerometer sensor and then compare those to the initial, base ESN classifier to see if adding the additional information improved the discriminative ability of the new classifier. Net acceleration power was chosen as the preferred derivative, as explained on page 206, because other researchers such as Z. He et al. (2008) and Kwapisz et al. (2011) have found it to be simple to calculate and independent of sensor orientation. The preference for sensor orientation independent data is further described in the Data section, Section 4.6.

The six iterations included V21, which simply added a single input vector to the ESN classifier that consisted of the raw net acceleration power values that ranged between zero and 1.732 and produced somewhat disappointing results with an area under the ROC curve of 91.99%, less than that produced by the initial, base classifier. Given the nature of the ESN $tanh$ transformation function, values close to and above 1.0 are difficult to differentiate and so some part of the net acceleration power's information is being lost. It is likely that this loss of information close to and above 1.0 at least partially accounts for this disappointing result although this is supposition rather than provable from this work. For the second iteration, V22, the raw net acceleration power values were divided by two to ensure that all input was between zero and one. This iteration produced a more useful result with an area under the ROC curve of 96.25%, greater than the 94.18% from the initial, base, real world ESN classifier.

With the third iteration, V23, a decision was made to split the net acceleration power into a low frequency and a high frequency component and to use those as two input vectors. Verplaetse (1996) suggests that the human hand and wrist have an expected frequency of movement of less than 8–12Hz. Maurer, Smailagic, Siewiorek, and Deisher (2006) suggest that daily activities such a walking and running have a maximum frequency of less than 20Hz however, more specifically in the case of the data from this work which does not happen to include any human running, Bouten et al. (1997) suggests that walking has a frequency between 0.85Hz and 5Hz. As a result, a decision was made to split the signal at 5Hz to have some sort of separation between expected body movements and expected wrist and arm movements. This was also convenient as it is halfway between 0hz and 10Hz sample rate for this data. However, other researchers may prefer to do more analysis before adopting a similar split. This ESN classifier produced output that with an area under the ROC curve of 97.31%, produced only four False Positives at the 0.6 cut-off level and was the best of the series of six iterations.

The remaining three iterations in this series used the same input data as V23 but increased the neuronal size and varied the meta-parameters for the ESN classifier. None of these three variations produced results as good as V23 and are otherwise unremarkable. It should be noted that no attempt was made to tune the meta-parameters nor to tune the size of the ESN models. Possible future work includes a more struc-

ture, experimental attempt to optimise the results from this combination of inputs and classifier. The graph of the output from this test of the V23 ESN classifier (Figure 6.32 on page 211) is remarkable in that the it is mostly subdued during the sequences when the base class is present while showing definite increase during all Mount sequences except for the third Mount (see subsequent disclosure). The periods during the base class sequences when the output is higher coincide with participant brushing their horse's coat where the arm action is similar to the arm action when mounting. The author thinks that it is remarkable that despite a relatively small amount of training data and only seeing seven Mounts during training that this classifier is able to recognise subsequent, previously unseen, mounts with this level of reliability and with so few False Positives.

Throughout this work the author has used a classifier output cut-off level of 0.6 to differentiate between a successfully classified class and one that was not successfully classified. This figure of 0.6 was chosen arbitrarily at the beginning and was kept throughout the work for consistency. However, there is no logic behind choosing a cut-off at this level. The output of the classifier is not a probability, it is simply a number that comes out of the classifier and while it is usually below one and it is related to the input which is bounded, it has no formal bound. As a result and as demonstrated by the ROC curve, the cut-off level could just as easily be set to any other figure. For example, setting the cut-off level to 0.39 would result in the V23 classifier correctly classifying all six stirrup mounts and would result in only 66 False Positives, all clustered around the "brushing" activity that can be differentiated using alternate techniques. With this is mind, the author considers that the v23 classifier has demonstrated that a classifier artefact has been developed which is capable of reliably classifying stirrup mounts within the data used in this work.

Disclosure: While reviewing the video data for the test files to ascertain what activities the participants were undertaking while the False Positives were generated the author also decided to review the third Mount that has consistently, throughout this real world work, failed to be correctly classified or has even been close to being classified in all cases except that with the strange ESN model meta-parameters encountered in the Ensemble section, Section 6.7. On review, embarrassingly, it was noted that this third Mount was not a stirrup mount like all the others and that it was instead an assisted Mount from a mounting block. While it is embarrassing to have not found this class error until this time, on the flip side, it demonstrates that the classifier is working even better than expected as with this Mount removed, all other Mounts have resulted in a substantial output up-tick that in coincidental with the Mount. While this is a net positive result for this classifier it does highlight that the classifier is quite specialised (by design) and that additional classifiers are required in order to get comprehensive classification across an area.

7.4.11 *Gyroscope and Accelerometer data with Lukosevicius code*

This design iteration is described in Section 6.12 on pages 224 through 231 and the results are discussed in Section 6.12.10 on page 231. This iteration is designed to demonstrate that the classification done within this work is generalisable to alternate

code bases. In this iteration a different code library is substituted for the library that has been used through out the prior work. Lukoševičius (2012) published ESN R code at http://212.201.49.24/sites/default/files/uploads/mantas/code/minimalESN.R. This code contained a small error that was corrected by the author. In this iteration the Lukoševičius code is presented with the same input data used with the V23 classifier.

The Lukoševičius code contains two major differences from the code used for the prior work, firstly, Lukoševičius's code discards the early output from ESN during training. When tuning the meta-parameters using a cost function based on the ESN output during training then discarding the early output as the reservoir is initially "filling up" provides a better comparison between ESN models and this is explained in Lukoševičius's publication. However, for this iteration where no meta-parameter tuning is being done this feature has no effect on the results. However, it is worth noting that attempts to classify any class when a ESN classifier is initialising its reservoir are unreliable. This has been ignored in this work because all testing is done on a pre-trained reservoir that is already initialised.

The second difference is that Lukoševičius's code has an extra input vector automatically built into it and this automatically generated vector always contains a continuous series of ones. Again, Lukoševičius (2012) explains the reasoning behind this. Lukoševičius explains that this is a more effective way of dealing with input data vectors that are symmetrically distributed around zero. Effectively, this is a better way of doing what was unsuccessfully attempted in Section 6.6. In addition, a decision was made to increase the neuronal count from 336 to 1,000 for this iteration as Lukoševičius's published work tends to use reservoirs of this size and this keeps his code consistent with his published work. this is despite the experience from the immediately prior work which produced slightly worse results with larger reservoirs. The other ESN meta-parameters are the same as used in the V23 iteration.

Two major differences can be noted from this iteration using Lukoševičius's code, firstly processing and run times increased more than 10 times from that of this project's code, reflecting that Lukoševičius's code has not been optimised for R. The second major difference is that the area under the ROC has increased even further to 99.70% (V23 97.32%) and with an increase to 39 (V23 22) True Positives at the 0.6 cut-off level. However, at the same cut-off level False Positives has increased to 38 (V23 4). This is reflected in Precision at 0.6 which has dropped from 0.84 for V23 to 0.51 for this iteration while Recall has increased from 0.19 for V23 to 0.35 for this iteration. looking at the output graph, Figure 6.45 on page 228 it can be seen that the output is very similar to that from V23 with the Mounts in the expected places and even the False Positives in the expected places. As expected from the area under the ROC curve value, the ROC is demonstrably better (see Figure 6.47a on page 230), especially at classifier output levels below 0.2.

Despite the differences in input and reservoir size the ESN classifier produced by Lukoševičius's code is similar enough to the v23 output that this design iteration can be said to have met it's design goals. As an added bonus, this iteration also suggests that adding an input vector of ones might be a simple way of further improving the V23 artefact. This is, however, supposition and has not been tested and is left for future work.

7.4.12 *Results Summary and Generalisation*

the results from iteration V23 demonstrate that the overall Design Science goal is complete and the artefact produced is capable of classifying the real-world riding data to the levels expected. In doing so, progress has not been linear but then that is the nature of the Design Science methodology. Through out the process areas have been identified as key to the eventual success and these ideas are discussed here so that the reader can better generalise this work. The ideas and areas are not necessarily discussed in the order in which they were presented within the work. The reader is reminded that while Design Science is a useful way of covering a lot of ground in a reasonable amount of time, one of its weaknesses is that almost none of the intermediate results are provable, only the successful delivery of the artefact at the end proves that it is possible to deliver an artefact on this nature. As such, the areas covered in this summary and generalisation are recommendations or heuristics rather than provable fact and while there is evidence to support the recommendations there is no experimental evidence. Indeed, the production of experimental evidence is something for future work.

At the beginning on this work the simplest possible data features were used for input, the raw data itself, after transforming it into a standardised -1 to +1 range. This was done with an eye to future implementation in an online, wearable situation. The simpler the data feature, the less processing is required to produce it and so this means that it can be produced faster than features that require more complex calculations. As the development proceeded slightly more complex features were added such as net acceleration power, however, these features were still simple to calculate. This preference for simpler features where ever possible when aiming for online classifiers to use within a wearable environment is supported by Plötz (2010).

Try simpler features first when aiming for online, wearable classifiers. Only add complex features if they are required.

Section 7.4.9 discussed using Butterworth filters and rolling means to detrend the Gyroscope data and to remove some level of high frequency noise. Doing this seems to have improved the performance of the artefact and it seems logical that consistent, noise free data would be easier to classify than data that contains drifts or non-random noise. Again, this approach is also recommended by Plötz (2010, p. 2). In this case a Butterworth filter was chosen to remove the drift because it is implementable in hardware, see Maji, Sree, Kar, Mandal, and Ghoshal (2015), and is therefore fast enough for online use. Similarly, a rolling mean was chosen to remove the high frequency noise because it is simple to implement in software with little processing.

Cleaning non-random and excessive noise and removing any drifts within the data before classification will tend to improve performance.

Section 7.4.10 discussed adding features from the accelerometer data to those from the Gyroscope data to improve performance. Other researchers including Maurer et al. (2006, p. 5) have also found that adding additional information from additional sensors resulted in improved classification performance and so this seems like a reasonable heuristic to recommend.

Adding additional information from additional sensors has improved classification performance in some circumstances

Section 7.4.8 discussed under sampling the majority class as one way of improving classification performance. In this work, under sampling seemed to provide mixed results. On one hand, the area under the ROC curve fell when compared with the initial, base, real world iteration that did not use aggressive under sampling but on the other hand the classifier output produced substantially less False Positives. In addition, there were other changes made between the initial, base, real world iteration and the iteration that used aggressive under sampling, especially having the meta-parameters tuned to the training data for the initial iteration and so exact comparisons are not possible and it is not possible to provably argue that the under sampling changes were beneficial. However, some positive results were noted and under sampling is recommended by a number of other researchers such as Drummond et al. (2003), Weiss (2004) and Chawla (2005) and so it seems reasonable to have the following heuristic.

In some cases under sampling of the majority class can improve classifier performance, possibly improving False Positive rates.

Section 7.4.7 discussed a simple ensemble classifier and as used within this work it did not improve classification performance however a number of authors including Chawla (2005), Z. Zhou and Liu (2006), Mollineda et al. (2007) and X. Liu et al. (2009) recommend ensemble classifiers, especially when working with unbalanced classes. In addition, Jaeger (2005) specifically recommends ensemble ESN classifiers. The author suspects that the chosen implementation technique within this work was not optimal and it is suggested that an ensemble more akin to that reported on by Koutnik et al. (2014) may be more effective, especially as a means of taking advantage of the capability of ESN classifiers to classify multiple classes in parallel.

The simple ensemble classifier that was tried within this work did not result in classification performance improvements, however, other researchers have reported success with ensemble classifiers including ESN ensemble classifiers and they are expected to have additional utility when used to classify multiple classes in parallel.

Section 7.4.6 discussed the use of an offset added to data that is symmetrical around a zero mean as a way of improving classifier performance, however, as implemented within this work this did not improve performance. In Section 7.4.11 however, a different way of improving classification performance for symmetrical data was tried as part of the Lukoševičius code iteration and this did seem to have positive results. In addition the addition of an input vector of all ones is highly recommended by both Jaeger (2005) and Lukoševičius (2012).

When using ESN classifiers several authors report achieving improved performance when an additional input vector of all ones is added

Finally, overall, this work has demonstrated that an ESN classifier was able to successfully classify a punctual activity from equestrian sport, Mounting, after reasonably minimal access to real world training data. This adds another tool for researchers looking at doing something similar.

Reservoir Computing techniques in general and Echo State Networks in particular have been used in this work to successfully classify a single, rare, punctual activity taken from equestrian sport. It is possible that similar techniques could be used by other researchers to classify other punctual activities.

The following sections discuss other areas that this work highlighted.

7.5 APPROACH

The approach taken within this research can be characterised as looking for a needle in a haystack. Basically the research tries to distinguish a relatively short and very rare sequence of movements from everything else (as the base class).

This work highlighted for the author the difficulties associated with classification of rare classes. This showed up in two key areas. Firstly, without some way of introducing a bias towards the class of interest into the model training process there was a tendency for the ESN model to be optimised for the much more common base class. During the third phase of development two techniques were used to try to bias the classification towards the class of interest. The first technique was to take out (under sample) some of the base class data from the training and tuning datasets (but never from the testing dataset). This had an added bonus of speeding up the model development and parameter tuning process as there was less data to deal with during these parts of the process. The second technique that was used to bias the model was to use a conditional mean square error as the cost function to be minimised during tuning rather than a straightforward mean square error cost function. Both of these techniques had some apparent advantages and this was demonstrated best in the later models where both the output from the ESN model showed a tendency to cluster closer to zero during base activities while maintaining a differential above zero for mount activities and by the increase in the area under the ROC curve.

In addition, within this work the ideal waveform can be characterised by a square wave that pops up from a sea of flatness. This makes attempts at optimising the output and/or computing (reducing) the error from the ideal waveform difficult as the very large base class dominates the results because of its size relative to the activity of interest (For example, typically one hour of riding or 3600 seconds of base class versus 5 seconds of mounting activity or more succinctly, 0.14% of activity versus 99.86% of base class). As a result of the dominance by the base class, there is a strong tendency to classify an instance as the base class by default. Of course, other researchers have had similar issues and there are established techniques available to mitigate some of the issues. Nevertheless, searching for rare data is considered to be a difficult task.

This isn't typically how the world occurs though. Mounting (or dismounting) a horse is not an activity that occurs against a background of no other activity or some homogeneous Null/Base activity. At each moment for which there is data, there is an activity occurring. Some of these other activities are punctual and quickly change while others are durative and persist for some period. An alternative then is not to distinguish one activity of interest from the base (nothingness) class but instead to

distinguish the activity of interest from other (perhaps similarly signatured) activities. When stated this way other possible approaches open up.

One important reason that many Activity Classification researchers follow the approach of distinguishing a particular activity from the Base/Null class is that manually classifying activities to provide a training signal is extremely time consuming and so spending the time to manually classify every activity occurring within the data stream is generally not attempted. Manually classifying the training signal does not scale up at all well and so most activity classification research only looks at a very small number of relatively common activities. In addition, in order to manually classify every possible activity within the data stream the researcher needs a taxonomy that covers every possible activity. Developing such a taxonomy of all possible activities is in itself a major task (maybe impossible to agree).

Another reason why most Activity Classification researchers set out to distinguish a single activity from a base class is that most of the more common tools for comparing one classification attempt against another (such as the area under the ROC curve) are designed to work on two–class problems (base class and one other class). While some three class and multi-class comparators exist they tend to be much less main stream and more complex to use. In addition, in situations similar to those followed in this research where a (machine learning) algorithm needs to have its parameters optimised in order to provide a "better" classifier then it becomes more difficult to optimise for two classes at the same time. This issue was one of the main reasons why this research changed from the initial intent of classifying both mounts and dismounts to only classifying mounts. It became too complex and time consuming to optimise the RC model for two classes at the same time. Optimising a classifier algorithm across more than two classes at the same time becomes even more difficult.

Distinguishing a single or small subset of activities from the Null/Base class is a potentially useful shorthand approach when activity and Null classes are relatively evenly matched (such as in laboratory situations) but hides a potentially more useful approach. Interestingly, durative activities tend to be less affected by this phenomena of the need to search for rare data than punctual activities, because by nature, durative activities persist for longer periods of time than punctual activities and so tend to be less rare.

When considered against a noisy background of activities occurring at each instant including activities, meta-activities and sub-activities at differing levels instead of against a quiet background of the Base activity, then it becomes clearer that a layered approach may be more useful. Rather than immediately trying to distinguish a particular activity of interest from everything else, it may be possible to (perhaps automatically) segment the data stream into "components" or motifs using some criteria. Say energetic components from not-so-energetic components, as a possible example. This segmentation and sub-segmentation would proceed to some depth (but not too deep). Wyatt, Philipose, and Choudhury (2005), Choudhury et al. (2008) and Vahdatpour et al. (2009) discuss similar concepts. Once the automatic pre-segmentation into motifs or components is done then a classifier is presented with the problem of distinguishing the particular activity of interest from other activities with similar characteristics from within the component segment that it occurs within.

Sifting the activity of interest from those other activities that are grossly different from it during the early stages of classification may result, in many cases, in a large part of the data stream able to be safely ignored when classifying this particular activity. The issue then becomes distinguishing the activity of interest from the smaller subset of activities that are similar to it (in ways defined by the earlier layer separators). At once there is both a smaller hay stack to search through and an opportunity to use a specialized algorithm or set of parameters that is optimised for distinguishing activities with similar characteristics that occur within the segment that has been chosen. This could be particularly useful in situations such as that encountered in this research where a RC classifier that requires its parameters to be tuned is used. It then becomes possible to tune the RC model based on the general characteristics of the sifted components and then use the RC model's capability to classify multiple activities in parallel.

Another way of reducing the size of the search hay stack in some cases is to use domain knowledge relating to how differing activities relate to each other. Some activities are relatively independent of each other while other activities have a dependency of some sort. This can be especially useful when one of the activities is a durative activity. For example, *running* and *looking at your watch* are two activities, one durative and the other punctual, that are relatively independent of each other. It is possible to be *running* and at the same time either *looking at your watch* or not *looking at your watch*. In addition, it is possible to be *looking at your watch* while *running* or not *running*. With relatively independent activities the relationship can not be used to reduce the search space but with dependent activities it may be possible in some cases to use the relationship to reduce the search space. Taking the example of two equestrian related activities, *mounting*/*mounted* and *cantering* (or *trotting, galloping or walking*) then it is not possible to be *cantering* on a horse without first being *mounted* on the horse and in the general course of events it is not possible for both to occur at the same time. For a less obvious but similar example from equestrian sport, consider the example of *brushing the horse* (durative) and *mounting* (punctual). In the general course of events it is not possible to be both *brushing the horse* and *mounting* at the same time.

These relationships between the durative and punctual activities can be used to reduce the search space, provided that it is relatively easy to classify the durative activities and in many cases it is easier to classify the durative activities precisely because of their regular, cyclic form over a reasonably long period of time. Once the durative activities are classified then where there is a mutually exclusive relationship between the durative activities and the punctual activity then those periods when the durative activities occur can be subtracted from the data stream and so only the (hopefully) shorter periods left over need to be searched for the punctual activity.

The manual data and video synchronisation process is an excellent if somewhat crude example of how the relationships between the durative and punctual activities can be used to reduce the search space. While synchronising the sensor data and video files the author would first view the sensor data at a relatively high level. When viewing a real world riding session from a zoomed out perspective it is relatively obvious when a person is on a moving horse as the relatively regular rhythm of the

horse's gait shows up. Using this crude sieve, the author would then expect the mount to occur relatively close to (and before) the horse's gait is detected and the dismount to occur close to (and after) the horse's gait stops. Of course this is somewhat simplistic as the horse may stand still at any time and the gait may not be unique, but it serves to illustrate the idea that by identifying the gait sequences and removing them it is possible to significantly reduce the data to be classified and possibly simplify the classification. The following two figures (7.1 and 7.2) illustrate how the the higher frequency signal associated with the durative horse gaits stand out quite well from other activities.

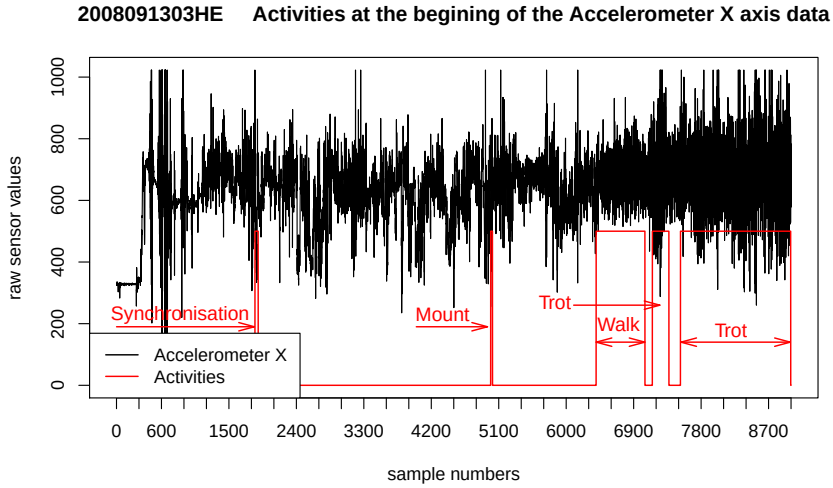


Figure 7.1: Example accelerometer X axis data stream illustrating relative ease of identifying periods when the horse gait is obvious

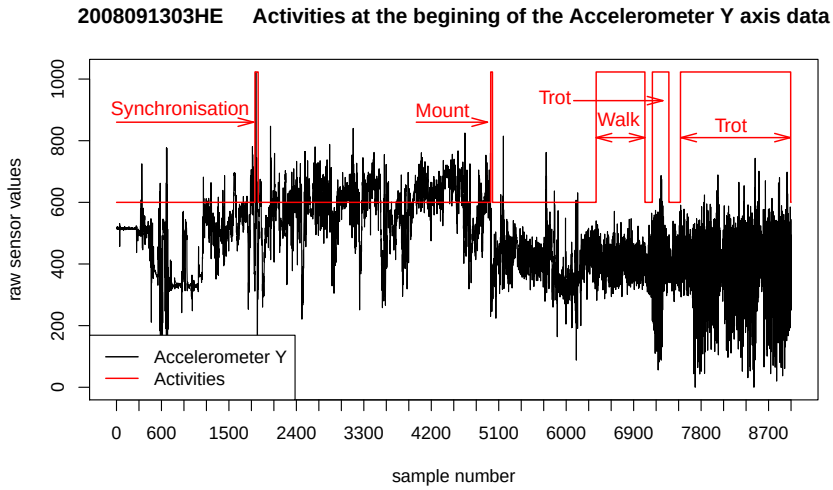


Figure 7.2: Example accelerometer Y axis data stream illustrating relative ease of identifying periods when the horse gait is obvious

7.6 SINGLE, WRIST MOUNTED SENSOR

The data used within this research was captured with the a single, wrist mounted IMU during Hunt (2009); this prior work will, for simplicity, be referred to as "prior work" from this point onwards. One of the key ideas from that prior work was to explore the usefulness of data from a single IMU during a period (2007–2008) when most published research was tending to use multiple IMUs.

During this research one of the strategies has been to favour simplicity (see Section 4.4.4). This is consistent with the use of a single data collection IMU and can be further observed in the tendency within the third design cycle of this research for the real–world based activity classification to only use the gyroscope data. However, as observed, during the final design cycle a derivative of the accelerometer data was added to the input with excellent effect. This additional discriminant power of the additional data suggests that adding more data may well be helpful to some extent. Of course, the extent of improvements from additional data has not been tested in this research and so it is impossible to say at what point the diminishing returns of extra discriminant power as opposed to the added complexity of dealing with the additional data will cross but it is likely that the cross over point of diminishing returns will be reached at some stage.

Leaving aside the untested suggestion of diminishing returns from additional data, there is at least one obvious hole with the data from the single IMU used to capture the data for this research. This hole is that all subjects whose data was used in this research wore the single IMU on their right wrist. During the data capture from the prior work some data was captured from subjects that wore the IMU on their left wrist but unfortunately none of the data from these sessions was usable for a number of different reasons. Using only a single IMU places an artificial constraint on activity classification and probably restricts the ability to successfully classify activities. While there is definitely utility in being able to classify activities based on the data collected from a single sensor, namely, the ease of use of only needing a single device that needs to be put on prior to riding, this may be outweighed by the potential benefits of the additional data from another IMU.

During analysis for the prior work there was some evidence that a subject that mounted whilst holding an whip resulted in changes to the way that they mounted that may have resulted in those mounts being misclassified. Unfortunately, the single occasion during the prior work when a subject mounted whilst holding a whip (in their right hand) there were (other) data issues that prevented this session from being used within this research and so this went untested. Nevertheless, there is a possibility that using an IMU on both wrists may allow mount classification to be generalised across the relatively common occasions when riders do mount with objects in their hand(s). This is, perhaps, an area for future research.

In addition, it seems likely that IMUs or sensors placed on other parts of a rider's body may provide useful information that enhances classification. Indeed, discussions involving riders during the data capture for the prior work about the purpose of the data capture resulted in a number of riders asking why a wrist mounted sensor

was being used to capture data for mounts when data from a pressure sensor on the saddle or something similar seemed to be a simpler solution.

Undoubtedly, there are many situations when using additional sensors that either sense different characteristics or that are placed in other body positions would add significant utility and would enable easier differentiation between similar activities. For example, if it is assumed that brushing a horse has some similarity in signal artefacts to mounting a horse (quick yaw movement of the wrist) then having, say, another inertial sensor mounted on the rider's waist would (perhaps) make it easier to distinguish mounting (when the body moves upwards) from brushing (when the body is more likely to be stable).

This research was conducted using data from a single, (right) wrist mounted sensor but there is no suggestion that this is the only or preferred situation with regard to numbers of sensors or sensor placement. This fits well with the overall vision for this research which envisages multiple sensors of different sorts on a number of places on a rider's body as illustrated in Figure 1.1.

7.7 DATA CAPTURED AT 10HZ

The data from the prior work was captured at 10Hz. During the work done within the prior work some simple experiments were done with the IMU set to capture data at higher rates, see (Hunt, 2009, pg. 87) for some details on this. This work established that for the IMU sensor used and combined with the need to transfer that data over a Bluetooth connection to a data logging device, the maximum reliable sample rate for the IMU was 100Hz. Of course, other sensors, possibly using different data transfer technologies may well allow higher sensor sample rates and so the maximum rate encountered during the prior work is in no way an overall maximum given different technologies.

This research has only used real sensor data from one source and all of this data was captured at 10Hz. As a result, this research is unable to provide guidance on what might be the best or even a good sample rate for data capture for classification in equestrian sport. This research does not provide any insights into the possibility of better classification from a more fine grained, higher sample rate and it is not possible to say if there are helpful signal artefacts that are obscured at 10Hz that might become more observable at a higher sampling rate. This may well be something that could be investigated in future research. However, in terms of the preprocessing of the data and in particular synchronisation of the data to the video of the activities then it would have been slightly easier if the data had of been captured at a rate closer to the video rate of 25Hz and perhaps even 50Hz would have made synchronising the data and video somewhat easier. During synchronisation, it was often difficult to match the peak acceleration (or deceleration) rates with the overhead clap exactly because the video frame that best captured the point of impact for the overhead clap was only one of three potential video frames that matched a single set of IMU sample values and there was no guarantee that the IMU sample set actually captures the peak expected from the clap. A sample rate of 25Hz would also not guarantee capturing the peak or even matching the video as the two signals could be offset from each

other even though both are recorded at 25Hz. A 50Hz sample rate for the IMU might have provided a more accurate synchronisation process.

7.8 ALTERNATE DATA SOURCES

As noted in Sections 7.6 and 7.7, the data available to use within this research has partly directed some of the design choices. The data collected during the prior work required significant effort to collect, pre-process and manually classify and as far as the author knows is the only equestrian sport based dataset of this nature and so it is a very useful resource and this research would not have been possible without it or something similar. However, any attempt at developing a new class of classification algorithm such as applying RC techniques to classify punctual activities from inertial data that relied on a single dataset would leave open the question of if the success of the algorithm was somehow tied to the dataset and/or its characteristics.

This research does not answer the question of if the ESN classification model used within it is somehow tied to the particular dataset used. Answering that question was never one of the goals of this research. It is however, a worthwhile question to consider now that the ESN model has been shown to work with the data used within this research. Turning the question around, it would be useful to know if a ESN model can be generalised to successfully classify either other punctual activities or the same activities based on different data?

The easiest of the two options would be to test an ESN classifier on other punctual activities from an existing open–source data repository such as, for example, the data at the University of California, Irvine, **Machine Learning** data repository at http://archive.ics.uci.edu/ml/datasets.html?sort=nameUp&view=list; the Carnegie Mellon University **Multi-Modal Activity** database at http://kitchen.cs.cmu.edu/main.php, the University of Southern California **Human Activity** dataset at http://sipi.usc.edu/HAD/USC-HAD.zip, the **Opportunity Project** datasets at http://www.opportunity-project.eu/node/48; the **UniMiB Smartphone-based Human Activity Recognition dataset - UniMiB SHAR** at http://www.sal.disco.unimib.it/technologies/unimib-shar/ or the **BoxLab** repository of Instrumented Living datasets at http://boxlab.wikispaces.com/List+of+Home+Datasets. BoxLab includes datasets from Georgia Tech (Aware Home), University of Essex (iSpace/iDorm), University of Virginia (Smarthouse), MIT (PlaceLab), Duke University (Smart Home) and University of Missouri (Tiger Place). Of these the University of California, Irvine collection seems to be the most comprehensive although many of the datasets it contains that contain inertial data for human activity are completely focussed on Durative activities and so are in most cases unusable for classifying Punctual activities. There are some datasets such as PAMAP2 Physical Activity Monitoring Data Set at https://archive.ics.uci.edu/ml/datasets/PAMAP2+Physical+Activity+Monitoring that contain some data that is relevant to Punctual activities. However, perhaps the best publicly available dataset for Punctual and Durative activities may well be either the **UniMiB SHAR** dataset as it contains commonly researched Durative activities such as walk and run along with some key Punctual activities such as “getting up”, “standing up” and a number of falls or the “*Gestures*” dataset from the Opportunity

Project where subjects were recorded opening drawers and cupboards and other similar punctual everyday home activities or the Carnegie Mellon Kitchen dataset where a number of subjects follow a recipe to cook a dish and the various punctual activities are noted.

Less easy but of more interest to the author would be to collect new equestrian sport related data using newer sensors such as the Yost Labs 3-Space sensors (see: https://www.yostlabs.com/yost-labs-3-space-sensors-low-latency-inertial-motion-capture-suits-and-sensors) which are capable of outputting Quaternions, Euler angles and rotation matrices as well as raw inertial readings. Other, similar, commercial sensors are also available.

Any such work, of course, would be future research. The ready availability of these data resources does however, give confidence that this research can be extended and developed further in a number of ways.

7.9 COMPARING RC TECHNIQUES AGAINST OTHER TECHNIQUES

This research does not claim that RC based activity classification techniques are better than other classification techniques nor even that RC based techniques are even comparable. This was never a goal of this research. The goal of this research was to design and construct an RC based technique to reliably classify a single punctual activity from Equestrian sport. This research makes no claim to generalise that ability to classify punctual activities beyond that single, specified activity.

Of course, having a classification technique that is only applicable to a single activity is somewhat limiting and so future research that extends the generality of RC techniques for classifying other punctual activities would significantly enhance the value of RC techniques in this area. Given the availability of other datasets that contain punctual activities as described in Section 7.8, it is easy to envisage how this research could be extended by applying RC punctual activity classification techniques to other activities and then comparing these results to more traditional activity classification techniques such as SVM.

However, as noted in Section 3.2.19 on page 55 RC based classifiers have a number of useful attributes including possible advantages in terms of reduced storage, reduced computing requirements and potentially earlier classification in addition to any potential classification advantages for Punctual activities. As a result, a simplistic comparison between a more common classifier such as a SVM would not give a definitive answer in this context. Such a comparison needs a well thought out and extensive series of tests against other existing classification techniques and that is almost another thesis in itself and well beyond the scope and design of this project.

The use of the design science methodology was based around the concept of producing and testing a useful artefact, in this case, the RC punctual activity classifier. In conventional scientific terms my hypothesis was that a RC classifier was a useful tool, for a number of reasons, to identify punctual events within this dataset and for which a possible, real-world implementation within a constrained resource, wearable context existed. This “engineering” and design problem approach and methodology is suited to the multiple potential benefits of a RC classifier. As pointed out above, there

is no intention to state that a RC has better or even comparable accuracy to alternate classifiers, any attempt to prove that this approach is a more accurate classification method for Punctual activities in general would need an extensive comparison with other classifiers, and this represents important future work, but represents an extension of the original thesis and so is currently outside the scope of this work. Further, a comparison of accuracy alone would not do adequate justice to this work if it ignored the additional potential beneficial features of RC based classifiers.

Fortunately, other researchers are also starting to explore the use of RC based classification engines for human activity recognition and this work is starting to answer the wider question of where this technique fits into the overall landscape of possible classification engines. (Palumbo et al., 2016) cites the author's own work in their paper which explores the fusion of inertial data and wireless location data as input into an ESN human activity classifier. The authors of that paper report that their ESN classification system was used in a classification competition against more traditional activity classification methods and performed comparably against the other methods. Importantly, these authors also identified similar requirements for use in a resource constrained wearable environment and also chose ESN techniques for their potential computational, storage and response advantages.

(Mici, Hinaut, & Wermter, 2016) report that the best results from their use of an ESN classification engine for long duration activities within the Cornell Activity Dataset 120 (CAD-120) collection of video data produced a best set of results where accuracy, precision and recall were all higher than previously reported results and where their average across a set of parameters produced comparable results to previously reported attempts from other researchers. They concluded that ESN were a more preferable classifier for this data.

(Gallicchio & Micheli, 2016) report on their use of an ESN based classifier for human "gestures" classification using video data from a Microsoft ® Kinect ® during the AALTD2016 challenge. In this case the gestures have some features in common with this author's definition of Punctual activities. The authors of the paper report that their solution using the ESN classifier produced the 5th best result out of 22 entries with 94.4% accuracy during the challenge blind test. They further reported that shortly after the challenge they introduced and ensemble ESN classifier that achieved an additional 2% gain in accuracy and suggested that additional tuning was likely to produce even better results. They also reported that part of their reasoning for choosing the ESN classifier was its relative computation efficiency compared with some alternative (but unspecified) other techniques.

(Basterrech & Ojha, 2016) report on their use of an ESN based classifier to classify human activity from a limited subset of data collected from three subjects. The authors specifically chose the ESN technique because of its ability to process temporal data and because of their perception that it is an efficient technique. While this paper concludes that ESN based classifiers have "very good accuracy" there is no attempt to contextualise this with other classification techniques.

Overall, while none of the recent work in this area has looked as extensively at Punctual activities as we would like and there is only one inertial data based example of the use of a RC classifier in a situation where the RC technique is directly

comparable with other techniques, nevertheless, the work by other researchers in this area strengthens our own proposition that RC based classifiers are a serious option to consider when classifying human activities from inertial sensor time-series data and an appropriate tool for the author to choose to use within this thesis. In addition, the limited results to date indicate that the accuracy of RC based classifiers is at least comparable to other currently popular human activity classification methods in some cases and may even have improved classification accuracy in other cases. In addition and as already noted within the Literature Review, (Niebles et al., 2010) strongly supports the author's supposition that Durative and Punctual activities require different classification algorithms in order to take advantage of the differing aspects of each type of activity.

7.10 ACTIVITIES WITH DIFFERENT TEMPORAL CHARACTERISTICS

One of the potentially beneficial characteristics of RC based activity classification techniques is the possibility that multiple activities can be classified in parallel from the same RC model. Where this characteristic can be utilised there is potentially a considerable saving in computation resources, however, as was discovered during this research when the initial goal was to classify two punctual activities, *Mounting* and *Dismounting*, there are two issues that tend to make it difficult to use this ability to classify multiple activities in parallel in real situations.

The first difficulty is that the parameters that define the detailed behaviour of the RC model need to be tuned so that the model is optimised for reliable classification. As the number of activities is extended beyond a single activity then the tuning of the RC model parameters becomes increasing more complex as the reliability of the classifier needs to be balanced across all activities. The ability to effectively use a common RC model to classify multiple activities in parallel is especially challenging when the temporal characteristics of the different activities differ by a significant amount. For example, if one activity typically takes 3 seconds to perform while the other activity takes 13 seconds to perform then it can be difficult to tune the parameters for reliable classification of both activities.

While RC models do not require their input data to be pre-segmented into separate windows the model characteristics (and parameters) do affect how much data is *remembered* by the model. The number of neurons places a maximum cap on how much past data can be stored within the model and so this is a gross limit. However, the number of neurons tends to only be a limit for either very small models or for activities that take place over quite long periods of time. Of course the rate at which the input data was sampled also has a major affect. As an example from the data used within this research which was sampled at 10Hz, a 3 second activity would need an absolute minimum of 30 neurons to reliably classify that activity with a 40-50 minimum neuron model size being recommended, while a 13 second activity would need an absolute minimum of 130 neurons for reliable classification with perhaps at least 180 neurons or more being recommended. In addition to the gross limit that comes from the number of neurons the other factor (parameter) that affects how much data the model can *remember* is the Leak Rate.

The second difficulty lies in the general lack of simple tools to measure the increase (or decrease) in classification performance beyond the two class problem. This means that in a three (or more) class problem it becomes difficult to tell if any change has improved the model's overall ability to classify the classes or not in most cases and can be especially challenging if the performance on one class has improved while the performance on another class has got worse.

One possible solution to the issue of tuning the model parameters to match the requirements for differing temporal periods is suggested in Koutnik et al. (2014). In this paper the authors suggest using a *Clockwork* RNN with multiple modules within the one model where each module is a standalone RNN within an overall network of modules with each module's parameters optimised for a particular temporal period (clock rate) rather than optimised for each activity. All modules then feed into a mechanism for choosing the strongest classification. This potentially offers a solution that allows multiple activities to be classified in parallel without complex parameter tuning but does not, of itself, also resolve the issue of being able to compare the reliability of of a multi-class classifier with another multi-class classifier. Again, while this looks like interesting research it is marked for future work.

7.11 SQUARE VERSUS RISING WAVE CLASSIFICATION

During this work the boundaries of the activities of interest have been marked using a square wave technique. That is, the base class is represented as a series of zeros, the activity as a series of ones and the ones are aligned with the activity boundaries. This meant that the training signal that was used to train the model(s) was a square wave with the peak straddling the activity (see Figure 7.3).

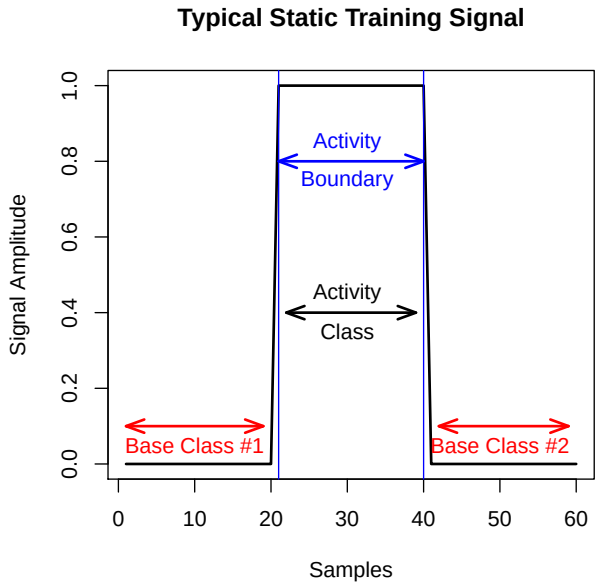


Figure 7.3: Typical Square Wave Training Signal

When this sort of training signal is used with a static, rather than a recurrent, classification technique such as SVM then both the data and the training signal are windowised and the ”window” is moved from left to right until it hopefully encapsulates enough of the data and training signal to ”learn” the patterns of the input data that falls within the activity boundaries. The training signal acts to bound the part of the input data that is of interest. In effect, for static, snapshot type techniques the activity boundary and the activity class are aligned.

With a recurrent classification technique such as ESN, the class boundary probably needs to move to the right and be offset from the the activity boundary because an ESN contains its own memory window (of some size) that can be thought of as trailing back to the left of the current cursor position. Considering Figure 7.3, when the ESN model cursor gets to 21 the classifier is asked to start outputting a one instead of a zero, however at this stage the classifier only has knowledge from the Base Class #1 area of input data (0–20).

If there is a pattern in the data within the activity boundaries then a recurrent classifier will only recognise that pattern once it is some way into the activity, perhaps it needs to be almost at 40 before it gets enough signal from within the activity bounds to correctly classify the activity. This behaviour is very typical of the results that were obtained during the design iterations and is demonstrated in Figure 7.4 which is taken from iteration 04-02.

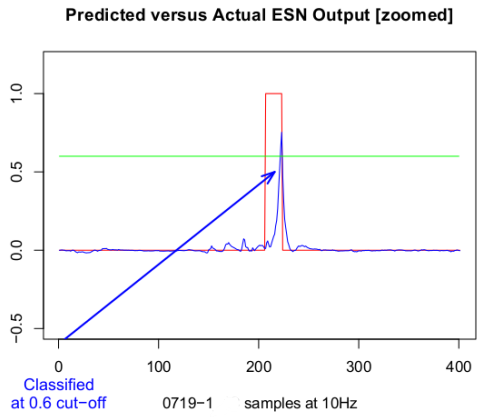
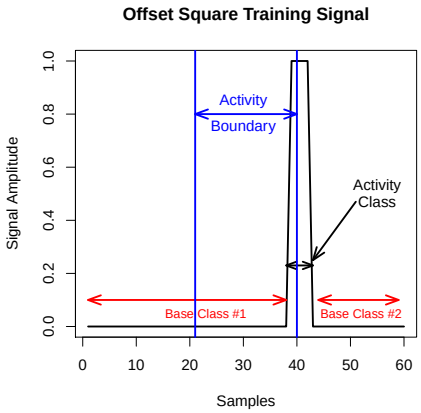
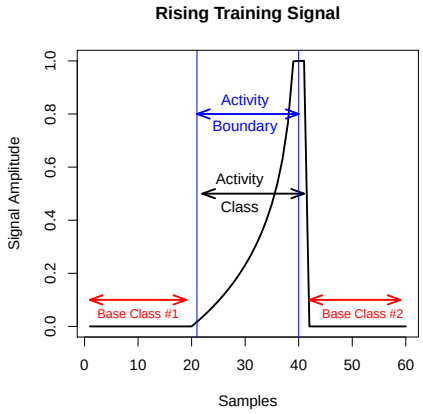


Figure 7.4: Exp. 04-02 0719-1 RB Mount

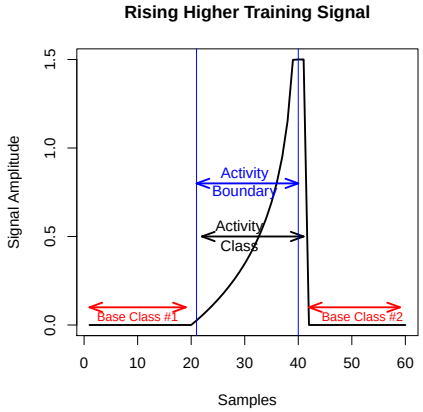
With this in mind, the author plans to investigate this idea as part of future work. Various forms of how the class boundary could be offset from the activity boundary include Figure 7.5a, which simply offsets the class while retaining its square shape and reducing its width, to Figures 7.5b and c where e a rising class signal is proposed that more closely matches the response that the model prefers to generate. The difference between the (b) and (c) versions is that the (c) version of the class signal rises to 1.5 rather than stopping at 1.0. The ESN model is designed to generate a signal as its output and so the training signal need not be factors or whole numbers or even be restricted to the limits that are applied when standardising the input data.



(a) Square Offset Class Signal



(b) Rising Class Signal



(c) Rising Higher Class Signal

Figure 7.5: Potential Alternate Class Signals

7.12 POSSIBLE IMPLEMENTATION ISSUES

The investigatory work done within this research has used large, pre-recorded sensor data files. A typical file, representing data from one multifaceted sensor, captured at only 10Hz over a typical riding session of 40 to 45 minutes represents around 20 megabytes of data. Other activity classification researchers have tended to use higher sampling rates to capture their data such as Moore, MacDougall, and Ondo (2008) (100Hz, ambulation), Chuang et al. (2012) (30Hz, exercise and ambulation), Novatchkov and Baca (2012) (100Hz, exercise and weight training), Qaisar et al. (2013) (150Hz, bowling in cricket) and Mitchell et al. (2013) (16Hz to 22Hz, general activities such as ambulation and sitting). As discussed in Section 7.7, If we implemented this technology with a higher sampling rate of, say, 100Hz while keeping everything else the same then we would increase the data size by one magnitude to around 200 megabytes for the same time interval. However, size issues don't end there. As envisaged, a wearable coaching system (see Section 1.2) would use multiple sensors on

multiple parts of an athlete's body and a training session may cover a much longer period than 40 to 45 minutes and so potentially data sizes could be a further magnitude larger at around 2 gigabytes or more per training session. Fortunately though, the classification engine used can be run on a step by step basis and so typically the main issue associated with computing with very large data files on lower powered wearable systems is a storage issue rather than a computational issue, although increasing data rates do require increases in compute cycles to enable the classification to keep up with the sample rate. Large storage requirements are still somewhat of an issue during data collection for classifier training however.

A relevant question for technocrats and researchers looking at implementing a RC based activity classifier similar to the one that this work has created is *How much computing power is required to implement a classifier such as this in a wearable device situation?* Of course, this is an impossible question to answer definitively as it is dependent on a number of issues, not the least of which is the level of computing power available on the chosen hardware platform. In a situation where the computing device is expected to be worn on the human body then normally computing power is constrained by weight, size and (battery) power available. However, some idea of a comparative answer is possible. In a general sense, Lukoševičius (2012) *A Practical Guide to Applying Echo State Networks* is a good place to start when considering applying an ESN.

One of the benefits of the ESN technique is that the highest compute intensive part, the tuning of the model, and the less compute intensive training process can be done offline before the model is used and so this means that only the classification part needs to be done as the activity is happening. Within this work the classification was done across an entire file but even so the compute requirements were relatively minor with the later iterations requiring, on average, 30 seconds of user compute time to classify the whole file. Table 6.42 provides an example. In addition, it is envisaged that the classifier would operate on a step–by–step mode, classifying each sample as it is received and although there is no empirical evidence from within this work to support this, it is possible that this level of classification would be achievable on a modern smartphone. Future work is planned to investigate this possibility.

7.13 LIMITATIONS

While it is natural for any author to want to highlight the usefulness and benefits of their work it can also be educational to point out the limitations. This section summarises and discusses some limitations of this work and the artefact that it produced.

7.13.1 *The role of the artefact within Wearable Coaching*

In Section 1.2 on page 3 the concept of a Wearable Coach was introduced and within this concept a number of necessary components were presented including a series of classifiers and from this the justification for this classifier artefact was drawn. Essentially, the process envisaged within this idea of a wearable coach was to use a series of classifiers to establish what activity a person (wanting coaching) is currently doing,

then to compare how they are doing that activity against some sort of template using a "comparator" and then if their technique is not ideal to start applying feedback until such time as they correct their technique. This can be (and was) broken down into separate components, namely classifiers, comparators, decision systems and feedback systems that then need to be managed so that they work together effectively using some sort of management module.

However, a coaching or training system need not be so disjoint. The idea of ideomotor feedback control has been around for some time and other researchers such as Galtier (2014) have not only demonstrated such systems but, as Galtier has done, have demonstrated them using RC techniques, ESN in Galtier's case. What ideomotor feedback systems are designed to do is to drive the output (the feedback) directly from the input. As an example, imagine an arm brace that was worn while drawing a circle. With an appropriate ideomotor feedback system then if when drawing the circle the arm started to move out of round then the feedback system would push it so that in went back into a round situation. Such a system encompasses all the components envisaged above into a single, consistent system. Using such technology within a wearable coaching system would make a classifier redundant and so this artefact, a stand-alone classifier, would be redundant. However, classifiers have a number of uses outside of wearable coaching as noted in Section 2.3.5 on page 20 and so producing this classifier artefact has value regardless of how a wearable coaching system is eventually implemented.

7.13.2 *Did some unique aspect of the Data shape the Artefact*

This work used a machine learning approach to build the classifier artefact and so, necessarily, the artefact itself is partially a product of the data with which it was trained. Of course, the classifier was then tested on similar data that it had not seen before and so there is some confidence that it is a generalised classifier that is not fundamentally restricted to the data used within this work. This is to be expected, however, this process leaves unanswered the question of if some, common, unique aspect of the data that was used has left a non-obvious influence on the artefact that would prevent the artefact from being used successfully with other data that does not have this unique aspect.

Firstly, the obvious artefacts are the data sample rate, the sensor's position and orientation on the wrist and the sensor itself. Clearly, a classifier trained on data samples at 10Hz would have unpredictable outcomes if presented with data sampled at a significantly different sample rate although I am unaware of any researcher who has actually tested this experimentally. It does however, seem to be, common sense.

With respect to sensor position and orientation on the wrist there is some confidence that the data is partially generalised and as a result the artefact is reasonably insensitive to minor differences in position and orientation. The data capture protocol required the subject to place the sensor on their own wrist and a deliberate decision was made to suggest only a general placement area for the sensor, although gross orientation was checked. This meant that participants were free to place and orientate the sensor where it was most comfortable, within reason. One participant did

attempt to place the sensor on the inside of their wrist but the researcher suggested to them that this might interfere with their riding and so it was moved to their outer wrist area. The deliberate policy of leaving participants to place the sensor themselves was done so as to best emulate what might happen in a real world situation where the researcher was not around to tell someone using the sensor how to place it. This meant that there was some variation both in placement and orientation of the sensor during data capture. Despite these changes in position and orientation (or perhaps even because of them) the classifier artefact has been able to generalise between each session dataset that has been included in the study. This ability to generalise was undoubtedly assisted by the features which were chosen as they were deliberately chosen to be relatively independent of orientation (e.g Gyroscope data and net acceleration power from the Accelerometer). Of course, gross changes in orientation such as a rotation in any plane much beyond around 20–30 degrees or more may well influence the Gyroscope data enough to prevent reliable classification but this has not been tested and so these comments are anecdotal only and should be treated with caution until tested. Gross changes in position are also likely to affect both the Gyroscope and Accelerometer data. As a result, the artefact is restricted to classifying data collected from a similar position and orientation to the data that was used for training purposes.

Hunt (2009, p.86) describes the sensor used. This sensor is no longer commercially available. Essentially the Accelerometer within the sensor collected acceleration data within a range of $\pm$ 2G's and the Gyroscope collected angular data with a sensitivity of 0 to 500 degrees per second. Both the Accelerometer and the Gyroscope collect analogue data that is then converted to a 32 bit real number using a 10 bit analogue-to-digital converter. Copies of the relevant data sheets which describe the individual components of the sensor, their recording range and expected accuracy can be found at Hunt (2009, p.179–182). Some inertial data based activity classification datasets such as the UniMiB SHAR dataset have been recorded using different sensors (smartphones) and the authors of that work, Micucci, Mobilio, and Napoletano (2016), showed that the sensors in each smartphone used were sufficiently similar that classification across sensors (smartphones) was not a problem. However, for this classification artefact it would not be safe to assume that data captured by different sensors with different specifications would be able to be used successfully without some attempt to standardise the data across the different sensors or to train a new classifier using data from both types of sensors. As a result, the classifier artefact as trained within this work is currently limited to working with data from a similar (no longer commercially available) sensor. this seems like a major limitation, however, the author contends that based on his experience building different iterations of the classifier artefact that the process of training the ESN classifier artefact is general enough that given different training data from a different sensor then an adequately performing classifier could be constructed to classify addition data from the new sensor. That is, while this instantiation of the artefact is sensor specific, it is relatively simple, now, to construct another classifier for a new sensor. This is a probable area for future research.

7.13.3 *No Guarantee of Implementation on a Wearable platform*

Section 3.2.21 on page 58 notes that implementing the resultant classifier artefact is out of scope for this work and so the fact that this classification artefact is not directly implementable on a wearable platform is not a limitation. However, it is reasonable to show that there is some possibility of implementing this classification artefact on a wearable platform. Jaeger (2003) showed that ESN can be implemented in such a way as to perform adaptive learning and on-line classification. Since then a number of authors including Jaeger and Haas (2004), Venayagamoorthy (2007) and Soh and Demiris (2014) have implemented ESN based classifiers that do on-line classification. From this it is possible to conclude that while the classification artefact from this work currently works on offline data it is possible to change it so that it also works on on-line data. However, that is only part of the picture as this classification artefact runs on a desktop computer with considerable processing power and memory resources whereas most wearable platforms have considerably less processing power and memory.

Prater (2016) suggests that the matrix multiplication that most implementations of ESN neural networks use when implemented on desktop computers are too resource intensive to use on wearable platforms such as Google's Android. This is probably so when ESN neural networks are implemented in high level languages such as R, as this work did but there is an open question about implementing an ESN neural network in lower level code on a wearable platform. Even with Prater's negative assessment of the ability to implement a standard ESN on Android, Prater did suggest a modified ESN solution using clustering techniques that does work on Android. In addition, smartphone and other wearable platforms continue to develop ever faster (multi) processors and greater memory resources and from this it is reasonable to assume that one way or another it would be possible to implement a modified version of the classifier artefact from this work on a wearable platform and this is suggested as possible future work.

7.13.4 *This classifier artefact only recognises a single activity*

During the initial stages of this work the author discovered that tuning the the global parameters that control the ESN configuration for two activities and reliably assessing classification accuracy improvements when two activities plus a base class was being considered meant this required much more work and was much more complicated than simply tuning for and measuring the improvement in a single activity. However, during later design iterations, at around Section 6.9 on page 190 it became apparent that the ESN meta-parameters did not need to be optimally tuned in order to achieve satisfactory classification rates. This is particularly apparent when looking at the note in Section 6.9.7 on page 192 and at Table 6.30 on page 192 and noting that the meta-parameters are (purposely) exactly the same as the meta-parameters used in the very initial real-world classification iteration.

The initial suggestion that every set of ESN meta-parameters needed to be individually tuned for each new activity would have meant that scaling from one classifying

one activity to classifying many activities would have effectively meant creating a new ESN for each activity and this would ultimately be untenable. However, what the later work within this project showed and what is backed up by authors such as Koutnik et al. (2014) and Lukoševičius (2012) is that a single reservoir and a single set of meta-parameters for that reservoir can classify multiple activities in parallel. The work of Koutnik et al., in particular, points to a technique using his "Clockwork RNN" where an ensemble of reservoirs set up to classify differing groups of activities that can be used to help scale across activities that would normally require quite diverse meta-parameters to work well.

While the work of Koutnik et al. (2014) suggests a way of scaling the ESN component there is still the issue that supervised training and hand classification for training and testing purposes can be extremely time consuming, as was discovered during this work and so scaling the training and testing data is still a big issue. Within certain constrained sporting domains such as equestrian sport it may be possible to to manually classify data and to continue to use supervised training but in less constrained domains other solutions are needed. Researchers such as Devert, Bredeche, and Schoenauer (2007) are pointing the way by suggesting techniques for unsupervised learning with ESNs and this is an important area for future work.

7.13.5 *This classifier artefact is limited to the Equestrian Sport Domain*

One of the key design decisions from the outset was to create a domain constrained activity classifier as explained in Section 1.9, page 10, Section 2.3.3 on pages 17 through 19 and in Section 4.4 and so it should be no surprise that the classifier artefact is indeed constrained to the domain of Equestrian sport. However, it is of some value to point out what this domain constraint means in a wider sense.

Based on the authors experience within Equestrian sport in New Zealand and his knowledge of the data collection process it is reasonable to state that the real world data used within this work is a reasonably good representation of many aspects of Equestrian sport. While the Swedish riders tend to ride more within indoor arenas as a result of their climate compared with New Zealand riders the data captured was from both situations and so the only real major difference between riding in these countries is covered and the author suspects that between the two countries the range of activities within European riding is well covered. This means that, but does not guarantee, that within the domain of Equestrian sport the classifier artefact is unlikely to classify some other equestrian related activity as a stirrup mount. There is no such confidence outside of the Equestrian sporting domain however.

It is quite possible that some other human activity in some other domain could produce a sensor signal that was close to or even the same as that produced by the mount activity. This is a limitation of all current classifiers as there is no universal activity classifier at this time and such a universal classifier would make further research into activity classification somewhat irrelevant. The author is not aware of any particular activity that produces a signal that might be easily confused with mounting a horse but one may exist and certainly within the data collected for this work it was found that brushing the horse produced a signal that initially was often confused with

mounting. A possibly more easy to understand example from a different domain may help in understanding. Within some of the more detailed data collected from instrumented homes such as the OPPORTUNITY dataset (Opportunity Consortium, 2016) there is signal data from activities such as shutting a drawer. It is possible that the data signal from a wrist mounted inertial sensor when closing a drawer may be very similar to a "thrust" motion when practising the sport of Fencing. Without some additional information from additional sensors it may be difficult to tell the two activities apart and so classification accuracy may well fall when the two activities co-occur without prior training for that eventuality. While all current activity classifiers suffer from this limitation to some level it is well to keep this limitation in mind when using this or any other classifier.

Chapter 8

FUTURE DIRECTIONS AND CONCLUSIONS

This chapter summarises the achievements of the presented research, provides several directions for future work and has some concluding remarks.

8.1 SUMMARY OF ACHIEVEMENTS

This work created a software artefact that successfully classifies the punctual activity *stirrup mounting a horse*. In doing this it has demonstrated that very short, punctual activities of one or two seconds duration and that occur only once can be distinguished from within complex, real world riding data of up to an hour and possibly longer. This is highly significant for technocrats seeking techniques for classifying other short duration and rare activities in other areas such as the researchers who are interested in auditing human activities in areas such as automotive assembly, health rehabilitation and other areas. More importantly in the sporting context within which this research commenced, it encourages coaches and athletes in equestrian and other sporting domains to consider how an automated system of auditing particular activities might help them train and compete more successfully. For example, the tennis coach that wants her pupil to include at least twenty backhand shots during each day's training session now has a potential way of automatically counting those shots provided a similar artefact can be developed and implemented in some sort of wearable device that was applicable to tennis. Even more specifically, this artefact if implemented into an equestrian orientated wearable device could automatically distinguish on–the–horse riding activities from ground activities and could thereby substantially improve the time taken to manually classify inertial data captured during a riding session for later analysis. An adaptation of this artefact may also be used to automatically classify on–the–horse activities so that there is less need for manual classification, thereby saving significant time and opening up the possibility of automatically analysing the on–the–horse activities.

8.2 FUTURE WORK

A number of possible future extensions to this work are possible and they fall into six broad areas, these are:

- Investigations of the usefulness of the distinction between Punctual and Durative activities when building activity classifies based on inertial data

- Investigations of the benefits of using RC techniques such as ESN and LSM as classifiers for Punctual and even Durative human activities
- Investigate the utility versus additional processing cost of simple features extracted from the data such as signal amplitude and acceleration power versus more complex features
- The implementation of the artefact into a situation where real world data needs to be classified in real-time as opposed to after the fact
- Building other components of the proposed wearable coaching system and integrating them into a wearable coaching device
- Other

While these broad areas are shown in a list it is not intended that they should necessarily be done in this or any other particular order although it is probably wise to further investigate the usefulness of the Punctual Vs Durative activity distinction reasonably early in the process. This may well be logically followed or paralleled by an investigation of RC based classification techniques versus other, more traditional techniques for both Punctual and Durative activities. This could then be followed and/or paralleled by the other work.

8.2.1 *Publicly Available Inertial Data Based Activity Datasets*

In almost all proposed areas of future work additional or alternative data will be required. There are a number of public, inertial data based, activity datasets that are available and these include:

- Human Activity Recognition (HAR) Using Smartphones, Anguita et al. (2013) - Durative daily activities only collected in a laboratory
- Quality of Life, De la Torre et al. (2009) - Punctual and Durative actions while following a cooking recipe in a laboratory
- UCI Machine Learning Repository, Lichman (2013) - A comprehensive collection of many datasets but most target meta activities with the data collected from instrumented homes
- A Falls dataset, Medrano, Igual, Plaza, and Castro (2014) - Only contains falls collected in a laboratory
- UniMiB SHAR, Micucci et al. (2016) - Large number of Durative and Punctual daily activities (getting up, standing up) plus falls collected in a laboratory
- OPPORTUNITY Datasets, Opportunity Consortium (2016) - Modes of locomotion (Durative) and, separately, Punctual activities such as open dishwasher, move cup and close drawer collected in a laboratory
- Exercise dataset, Reiss and Stricker (2012) - Sports orientated exercise activities collected in a laboratory
- BOXLab, Tangient LLC (2016) - Various datasets mostly aimed at meta activities and collected from instrumented homes

- Amsterdam Instrumented Home, van Kasteren, Englebienne, and Krose (2011) - A dataset aimed at meta activities collected from an instrumented home
- Mobifall, Vavoulas, Pediaditis, Spanakis, and Tsiknakis (2013) - Durative daily activities plus falls and two punctual activities, getting into and out of a car collected in a laboratory
- B-WAR, A. Y. Yang, Jafari, Sastry, and Bajcsy (2009) - Durative only daily activities collected in a laboratory
- USC-HAD, M. Zhang and Sawchuk (2012) - Durative only daily activities collected in a laboratory

However, perhaps the most interesting publicly available datasets for Punctual and Durative activities, from the author's perspective may well be either the UniMiB SHAR dataset as it contains commonly researched Durative activities such as walk and run along with some key Punctual activities such as "getting up", "standing up" and a number of falls; the "OPPORTUNITY" datasets from the Opportunity Project where subjects were recorded opening drawers and cupboards and other similar punctual everyday home activities or the Quality of Life dataset where a number of subjects follow a recipe and the various Durative and Punctual activities are noted.

Unfortunately, almost all of the publicly available datasets had the data collected in a laboratory situation and the closest that some of these datasets come to real world situations are when the data is collected in instrumented homes. This is in all probability due to the huge cost in terms of time that is needed to collect data in real life situations and probably also has some association with the privacy issues that relate to recording possibly intimate data from individuals in uncontrolled situations. This, in turn, highlights the value and uniqueness of the real world riding data that was used in this work.

8.2.2 *Investigate the distinction between Punctual and Durative activities*

The distinction between Punctual and Durative activities was highlighted within this work, based on similar activity classification using video data and based on anecdotal reporting from other researchers. However no attempt was made within this work to experimentally challenge this distinction. One way of challenging this distinction would be to take a dataset such as UniMiB SHAR dataset that contains both Durative and Punctual activities and experimentally testing a single classifier that has been trained to classify all activities against two specialised classifiers, one trained to classify Punctual and the other trained to classify Durative activities. Such an experiment would need to use the same classification engine (e.g. SVM, ESN, Random Forest) and the same features across both situations. For completeness, it would probably be necessary to test multiple classification engines and multiple feature sets to account for possible classification bias attached to either the classification engine or the chosen feature set.

8.2.3 *Investigate the benefits of Reservoir Computing classifiers vs other classifiers*

Perhaps the most obvious area of future work would be to compare the ability of RC techniques for classifying the punctual activity of interest in this publication with other, more established classification techniques, as described in section 7.9 on page 259.

This work could either be done by applying more established techniques, perhaps SVM based techniques to the existing equestrian data for a direct comparison with this work or alternatively by applying RC techniques to publicly available activity data such as the Punctual activity data from the UniMiB SHAR, OPPORTUNITY or Quality of Life datasets. The author's interest is towards the Equestrian data but this is a unique dataset and it could be argued that any results that come out of the work done using this dataset may be unique to the dataset and so it would probably be more productive to apply a RC classification engine to data such as that above that contains Punctual activities and which has already been classified by one or more other researchers using more traditional classification engines.

In addition, a potential project could look at the utility of RC based classifiers versus more traditional classifiers that exploit temporal aspects of the input signal such HMM based classifiers. In this case, activities need not be restricted to only classifying Punctual activities. The simplest project to set up would would use the UniMib SHAR dataset or something similar that contains both Durative and Punctual activities and which has previously been classified with more traditional classification engines. In considering this work it may be helpful to consider the work done by Basterrech and Ojha (2016) who very recently looked at applying ESN technology to classify Durative activities.

An area of future work that is close to the author's own personal interests would be to collect additional equestrian data, perhaps at a higher than 10Hz sample rate, as described in section 7.7 on page 257 and also perhaps utilising more than one sensor, as described in section 7.6 on page 256. In both these cases, a modern IMU sensor would be used that did not have some of the same drawbacks as the original sensor. With this data, collected at higher sample rates, it would be possible to apply more traditional classification engines to the data and to look at classifying both Punctual and Durative equestrian activities such as the horse's gait.

8.2.4 *Investigate the cost benefits of simple activity features Vs complex features*

A number of other authors (Banos, Damas, Pomares, Prieto, & Rojas, 2012; Gupta & Dallas, 2014; Pirttikangas, Fujinami, & Nakajima, 2006) have reported on work that analyses possible optimum feature sets for human activity classification, however, all of these works have looked at Durative activities only. This means that another possible avenue for future work is to do similar feature optimisation work on various Punctual activity datasets. In addition and with a view to implementing activity classifiers on wearable devices such as Smart Phones another possible area of future work is to look at the benefits to classification of additional information provided by more complex features versus the additional computing costs of calculating those features.

8.2.5 *Implementing the artefact in a real-time classification environment*

This work successfully built a classification artefact that runs off-line using a high-end desktop computer. The ultimate aim is to build a wearable system that operates in real-time to classify sensor data as it is received. Jaeger (2003) demonstrated that ESN are capable of both on-line classification and adaptive learning. What is missing and is a potential area of future work is to implement a RC classifier, probably one based on ESN technology because LSM technology generally requires more computational power than ESN (see Verstraeten et al., 2007) in either a smartphone platform or some sort of other wearable platform in a configuration that would allow for on-line classification. Prater (2016) suggests that ESN are too complex to implement on a smartphone platform but does not offer any evidence to back up this claim. However, Prater does offer a modification to the standard ESN based on clustering rather than output weight training which is said to decrease computational requirements and therefore, make it easier to implement an ESN classifier on a smartphone.

8.2.6 *Building other components of a Wearable Coach*

One of the theoretical benefits of RC classification techniques is the ability to classify multiple activity classes in parallel. This benefit was not utilised within the current work for a number of reasons including difficulties encountered trying to tune the RC parameters for differing activity attributes, especially differing temporal attributes. However, the author's experience during the later iterations confirmed that once RC model parameters were "good enough" in a generalised sense then reliable classification can be achieved without having to tune the RC model parameters for an exact "best match" situation. This then suggests that an area of future work could involve clustering activities with similar temporal characteristics together with good enough RC model parameters for the whole cluster might enable successful classification in parallel. This could be extended even further by utilizing the ideas expressed in Koutnik et al.'s *Clockwork RNN* in Koutnik et al. (2014). This possible area of future work is described in more detail in section 7.10, starting on page 261.

8.2.7 *Other possible areas of future work*

A more novel area for possible future work is the exploration of the effects of different classification wave forms as described in section 7.11 on page 262. An improvement in classification in this area may have wide application for other, non-RC classification techniques, as well as for RC techniques.

Section 7.5 on page 252 highlighted one of the key constraints that underlies most of the activity classification work, that manually labelling activities for machine learning and other similar, training based, techniques is very time consuming and so dealing with this issue is a general area of future work. Possible future work could include investigating techniques of automatically labelling activities using ideas from work such as Choudhury et al. (2008); Spriggs, De La Torre, and Hebert (2009); Vahdatpour et al. (2009); Wyatt et al. (2005).

8.3 CONCLUDING REMARKS

The research has met its design goals of producing a software artefact that adequately classifies the punctual activity *mounting a horse using a stirrup mount*. In the process it has highlighted some of the potential benefits of differentiating activities into punctual and durative meta-classes. In addition a number of areas of potential future research have been identified, demonstrating that this research has the potential to be added to in a number of ways.

REFERENCES

References marked with an asterisk indicate studies included in the meta-analysis.

Aarts, E., & Wichert, R. (2009). Ambient intelligence. In H.-J. Bullinger (Ed.), *Technology guide* (pp. 244–249). Springer Berlin Heidelberg.

*Abdallah, Z. S., Gaber, M. M., Srinivasan, B., & Krishnaswamy, S. (2015). Adaptive mobile activity recognition system with evolving data streams. *Neurocomputing*, *150*, 304–317. doi: 10.1016/j.neucom.2014.09.074

Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., & Steggles, P. (1999). Towards a better understanding of context and context-awareness. In *Handheld and ubiquitous computing* (pp. 304–307).

Aggarwal, J. K., & Cai, Q. (1997). Human motion analysis: a review. In *IEEE nonrigid and articulated motion workshop, 1997. proceedings* (pp. 90–102). doi: 10.1109/NAMW.1997.609859

Ahmadi, A., Rowlands, D., & James, D. A. (2009). Towards a wearable device for skill assessment and skill acquisition of a tennis player during the first serve. *Sports Technology*, *2*(3-4), 129–136. Retrieved 2014-06-03, from http://onlinelibrary.wiley.com/doi/10.1002/jst.112/abstract doi: 10.1002/jst.112

Alemdar, H. (2014). Multi-resident human behaviour identification in ambient assisted living environments. In *Icmi '14: Proceedings of the 16th international conference on multimodal interaction* (pp. 369–373). ACM Press. Retrieved 2015-01-27, from http://dl.acm.org/citation.cfm?doid=2663204.2666286 doi: 10.1145/2663204.2666286

*Allen, F. R., Ambikairajah, E., Lovell, N. H., & Celler, B. G. (2006). Classification of a known sequence of motions and postures from accelerometry data using adapted Gaussian mixture models. *Physiological Measurement*, *27*(10), 935.

*Alshurafa, N., Xu, W., Liu, J. J., Huang, M., Mortazavi, B., Sarrafzadeh, M., & Roberts, C. (2013). Robust human intensity-varying activity recognition using stochastic approximation in wearable sensors. In *Body sensor networks (BSN), 2013 IEEE international conference on* (pp. 1–6). IEEE.

*Altun, K., & Barshan, B. (2010). Human Activity Recognition Using Inertial/Magnetic Sensor Units. In A. A. Salah, T. Gevers, N. Sebe, & A. Vinciarelli (Eds.), *Human Behavior Understanding* (pp. 38–51). Springer Berlin Heidelberg.

*Altun, K., Barshan, B., & Tuncel, O. (2010). Comparative study on classifying human activities with miniature inertial and magnetic sensors. *Pattern Recognition*, *43*(10), 3605–3620. doi: 10.1016/j.patcog.2010.04.019

*Álvarez de la Concepción, M. A., Soria Morillo, L. M., Gonzalez-Abril, L., & Ortega Ramírez, J. A. (2014). Discrete techniques applied to low-energy mobile human activity recognition. a new approach. *Expert Systems with Applications*, *41*(14), 6138–6146. doi: 10.1016/j.eswa.2014.04.018

Amer, M. R., Xie, D., Zhao, M., Todorovic, S., & Zhu, S. (2012). Cost-sensitive top-down/bottom-up inference for multiscale activity recognition. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, & C. Schmid (Eds.), *Computer vision Ű ECCV 2012* (Vol. 7575, pp. 187–200). Springer Berlin Heidelberg.

Amft, O., Amstutz, R., Smailagic, A., Siewiorek, D., & Troster, G. (2009). Gesture-controlled user input to complete questionnaires on wrist-worn watches. In *Human-Computer Interaction. Novel Interaction Methods and Techniques* (pp. 131–140). Springer. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-02577-8_15

*Amft, O., Junker, H., & Troster, G. (2005). Detection of eating and drinking arm gestures using inertial body-worn sensors. In *Wearable computers, 2005. proceedings. ninth ieee international symposium on* (pp. 160–163).

Amft, O., Lombriser, C., Stiefmeier, T., & Troster, G. (2007). Recognition of user activity sequences using distributed event detection. In *Smart sensing and context* (pp. 126–141). Springer.

*Aminian, K., Najafi, B., Bula, C., Leyvraz, P. F., & Robert, P. (2002). Spatio-temporal parameters of gait measured by an ambulatory system using miniature gyroscopes. *Journal of biomechanics*, *35*(5), 689–699. Retrieved 2015-06-09, from http://www.sciencedirect.com/science/article/pii/S0021929002000088

*Aminian, K., Rezakhanlou, K., De Andres, E., Fritsch, C., Leyvraz, P. F., & Robert, P. (1999). Temporal feature estimation during walking using miniature accelerometers: an analysis of gait improvement after hip arthroplasty. *Medical & biological engineering & computing*, *37*(6), 686–691. Retrieved 2015-06-09, from http://link.springer.com/article/10.1007/BF02513368

*Aminian, K., Robert, P., Buchser, E. E., Rutschmann, B., Hayoz, D., & Depairon, M. (1999). Physical activity monitoring based on accelerometry: validation and comparison with video observation. *Medical & Biological Engineering & Computing*, *37*(3), 304–308. Retrieved 2015-06-03, from http://link.springer.com/article/10.1007/BF02513304 doi: 10.1007/BF02513304

*Aminian, K., Robert, P., Jéquier, E., & Schutz, Y. (1995). Incline, speed, and distance assessment during unconstrained walking. *Medicine and science in sports and exercise*, *27*(2), 226–234. Retrieved 2015-06-09, from http://europepmc.org/abstract/med/7723646

*Amma, C., Gehrig, D., & Schultz, T. (2010). Airwriting recognition using wearable motion sensors. In *Proceedings of the 1st Augmented Human International Conference* (p. 10). ACM.

*Anguita, D., Ghio, A., Oneto, L., Parra, X., & Reyes-Ortiz, J. L. (2013). A Public Domain Dataset for Human Activity Recognition Using Smartphones. In *21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013* (p. 6). Bruges, Belgium.

Asadzadeh, P., Kulik, L., & Tanin, E. (2012). Gesture recognition using RFID technology. *Personal and Ubiquitous Computing*, *16*(3), 225–234. Retrieved 2015-01-27, from http://link.springer.com/10.1007/s00779-011-0395-z doi: 10.1007/s00779-011-0395-z

*Atallah, L., Lo, B., King, R., & Yang, G. (2011). Sensor Positioning for Activity

Recognition Using Wearable Accelerometers. *IEEE Transactions on Biomedical Circuits and Systems*, *5*(4), 320–329. doi: 10.1109/TBCAS.2011.2160540

Avci, A., Bosch, S., Marin-Perianu, M., Marin-Perianu, R., & Havinga, P. (2010). Activity Recognition Using Inertial Sensing for Healthcare, Wellbeing and Sports Applications: A Survey. In *2010 23rd international conference on architecture of computing systems (ARCS)* (pp. 1–10).

Baca, A. (2012). Methods for recognition and classification of human motion patterns - A prerequisite for intelligent devices assisting in sports activities. In *MATHMOD 2012, proceedings* (Vol. 14). Vienna, Austria. Retrieved 2014-06-03, from http://seth.asc.tuwien.ac.at/proc12/full_paper/Contribution438.pdf

Baca, A., Dabnichki, P., Heller, M., & Kornfeind, P. (2009). Ubiquitous computing in sports: A review and analysis. *Journal of Sports Sciences*, *27*(12), 1335–1346. Retrieved 2015-01-21, from http://www.tandfonline.com/doi/abs/10.1080/02640410903277427 doi: 10.1080/02640410903277427

Bachlin, M., Forster, K., & Troster, G. (2009). SwimMaster: A wearable assistant for swimmer. In *Proceedings of the 11th international conference on ubiquitous computing* (pp. 215–224). New York, NY, USA: ACM. doi: 10.1145/1620545.1620578

Bachlin, M., & Troster, G. (2011). Swimming performance and technique evaluation with wearable acceleration sensors. *Pervasive and Mobile Computing*, *8*, 68–81. Retrieved 2011-07-18, from http://www.sciencedirect.com/science/article/pii/S157411921100071X doi: 10.1016/j.pmcj.2011.05.003

Bachmann, E. R. (2000). *Inertial and Magnetic Tracking of Limb Segment Orientation for Inserting Humans into Synthetic Environments* (Unpublished doctoral dissertation). Naval Postgraduate School, Monterey, California.

Bachmann, E. R., Duman, I., Usta, U. Y., McGhee, R. B., Yun, X. P., & Zyda, M. J. (1999). Orientation tracking for humans and robots using inertial sensors. In *Computational intelligence in robotics and automation, 1999. cira'99. proceedings. 1999 ieee international symposium on* (pp. 187–194).

Bacic, B. (2004). Towards a neuro fuzzy tennis coach: Automated extraction of the region of interest (ROI). In *Fuzzy systems, 2004. proceedings. 2004 IEEE international conference on* (Vol. 2, pp. 703–708). IEEE. Retrieved 2014-06-03, from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1375484

Back, T., & Schwefel, H. (1993). An overview of evolutionary algorithms for parameter optimization. *Evolutionary computation*, *1*(1), 1–23.

*Bannach, D., Lukowicz, P., & Amft, O. (2008). Rapid prototyping of activity recognition applications. *Pervasive Computing, IEEE*, *7*(2), 22–31.

Banos, O., Damas, M., Pomares, H., Prieto, A., & Rojas, I. (2012). Daily living activity recognition based on statistical feature quality group selection. *Expert Systems with Applications*, *39*(9), 8013–8021. doi: 10.1016/j.eswa.2012.01.164

*Baños, O., Damas, M., Pomares, H., & Rojas, I. (2013). Activity recognition based on a multi-sensor meta-classifier. In *International work-conference on artificial neural networks* (pp. 208–215). Springer.

*Bao, L., & Intille, S. (2004). Activity recognition from user-annotated accelera-

tion data. In A. Ferscha & F. Mattern (Eds.), *Pervasive computing* (pp. 1–17). Springer Berlin Heidelberg.

Barbic, J., Safonova, A., Pan, J. Y., Faloutsos, C., Hodgins, J. K., & Pollard, N. S. (2004). Segmenting motion capture data into distinct behaviors. In *Graphics interface, proceedings of. 2004* (pp. 185–194). Waterloo, Ontario, Canada: Canadian Human-Computer Communications Society. Retrieved 2011-07-29, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.59.376&rep=rep1&type=pdf

Basterrech, S., & Ojha, V. K. (2016). Temporal learning using echo state network for human activity recognition. In *Proceedings of the third ieee european network intelligence conference* (pp. 217 – 223). doi: 10.1109/ENIC.2016.039

*Bayat, A., Pomplun, M., & Tran, D. A. (2014). A study on human activity recognition using accelerometer data from smartphones. *Procedia Computer Science*, *34*, 450–457. doi: 10.1016/j.procs.2014.07.009

Benbasat, A., & Paradiso, J. (2002). An inertial measurement framework for gesture recognition and applications. In *Gesture and sign language in human-computer interaction* (pp. 9–20). Springer.

Benbasat, A. Y. (2000). *An inertial measurement unit for user interfaces* (Unpublished doctoral dissertation). Citeseer.

Bendtsen., C. (2012). pso: Particle swarm optimization [Computer software manual].

*Berchtold, M., Budde, M., Schmidtke, H. R., & Beigl, M. (2010). An extensible modular recognition concept that makes activity recognition practical. In *KI 2010: Advances in Artificial Intelligence* (pp. 400–409). Springer. Retrieved 2015-06-05, from http://link.springer.com/chapter/10.1007/978-3-642-16111-7_46

Bernmark, E., & Wiktorin, C. (2002). A triaxial accelerometer for measuring arm movements. *Applied Ergonomics*, *33*(6), 541–7.

Biswas, K. K., & Basu, S. K. (2011). Gesture recognition using microsoft kinect. In *Automation, robotics and applications (ICARA), 2011 5th international conference on* (pp. 100–103). IEEE. Retrieved 2015-01-27, from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6144864

*Blum, M. L. (2005). *Real-time Context Recognition* (Unpublished doctoral dissertation). Swiss Federal Institute of Technology, Zurich.

Bobick, A. F. (1997). Movement, activity and action: the role of knowledge in the perception of motion. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *352*(1358), 1257–1265. Retrieved from http://rstb.royalsocietypublishing.org/content/352/1358/1257 doi: 10.1098/rstb.1997.0108

Bohte, S. M., Kok, J. N., & Poutré, J. A. L. (2002). Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing*, *48*(1–4), 17–37.

*Boissy, P., Choquette, S., Hamel, M., & Noury, N. (2007). User-Based Motion Sensing and Fuzzy Logic for Automated Fall Detection in Older Adults. *Telemedicine and e-Health*, *13*(6), 683–694. Retrieved 2015-06-10, from http://online.liebertpub.com/doi/abs/10.1089/tmj.2007.0007 doi: 10.1089/

tmj.2007.0007

*Bosch, S., Marin-Perianu, M., Marin-Perianu, R., Havinga, P., & Hermens, H. (2009). Keep on Moving! Activity Monitoring and Stimulation Using Wireless Sensor Networks. In P. Barnaghi, K. Moessner, M. Presser, & S. Meissner (Eds.), *Smart Sensing and Context* (pp. 11–23). Springer Berlin Heidelberg. Retrieved 2015-06-03, from http://link.springer.com/chapter/10.1007/978-3-642-04471-7_2

*Bourke, A., OBrien, J., & Lyons, G. (2007). Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. *Gait & Posture*, *26*(2), 194–199. Retrieved 2015-06-10, from http://linkinghub.elsevier.com/retrieve/pii/S0966636206001895 doi: 10.1016/j.gaitpost.2006.09.012

*Bourke, A., ODonovan, K., & OLaighin, G. (2008). The identification of vertical velocity profiles using an inertial sensor to investigate pre-impact detection of falls. *Medical Engineering & Physics*, *30*(7), 937–946. Retrieved 2015-06-10, from http://linkinghub.elsevier.com/retrieve/pii/S1350453307002081 doi: 10.1016/j.medengphy.2007.12.003

*Bouten, C. V., Koekkoek, K., Verduin, M., Kodde, R., & Janssen, J. D. (1997). A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity. *Biomedical Engineering, IEEE Transactions on*, *44*(3), 136–147. Retrieved 2015-06-12, from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=554760

Bowen, J., Hinze, A., & Cunningham, S. J. (2015). Into the woods. In *INTERACT 2015 adjunct proceedings: 15th IFIP TC. 13 international conference on human-computer interaction 14-18 september 2015, bamberg, germany* (Vol. 22, p. 137). University of Bamberg Press.

*Boyle, J., Karunanithi, M., Wark, T., Chan, W., & Colavitti, C. (2006). Quantifying functional mobility progress for chronic disease management. In *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE* (pp. 5916–5919). IEEE. Retrieved 2015-06-12, from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4463154

Bradley, A. P. (1997, July). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, *30*(7), 1145–1159. doi: 10.1016/S0031-3203(96)00142-2

Brand, M., Oliver, N., & Pentland, A. (1997). Coupled hidden markov models for complex action recognition. In *Computer vision and pattern recognition, 1997. proceedings., 1997 IEEE computer society conference on* (pp. 994–999). IEEE. Retrieved 2015-01-28, from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=609450

*Brezmes, T., Gorricho, J.-L., & Cotrina, J. (2009). Activity Recognition from Accelerometer Data on a Mobile Phone. In S. Omatu et al. (Eds.), *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living* (pp. 796–799). Springer Berlin Heidelberg. Retrieved 2015-06-05, from http://link.springer.com/chapter/10.1007/978-3-642-02481-8_120

Brodie, M., Walmsley, A., & Page, W. (2008). Fusion motion capture: a proto-

type system using inertial measurement units and GPS for the biomechanical analysis of ski racing. *Sports Technology, Journal of*, *1*(1), 17–28.

Brunetti, F., Moreno, J. C., Ruiz, A. F., Rocon, E., & Pons, J. L. (2006). A new platform based on IEEE802. 15.4 wireless inertial sensors for motion caption and assessment. In *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE.* Institute of Electrical and Electronics Engineers. Retrieved from http://digital.csic.es/handle/10261/43640

*Brusey, J., Rednic, R., & Gaura, E. (2009). Classifying Transition Behaviour in Postural Activity Monitoring. *Sensors & Transducers*, *7*, 213–223.

Brush, A., Krumm, J., & Scott, J. (2010). Activity recognition research: The good, the bad, and the future. In *Pervasive 2010 Workshop.* Helsinki, Finland. Retrieved 2013-08-10, from http://ubicomp.lancs.ac.uk/workshops/activity2010/Brush_Krumm_Scott_Microsoft.pdf

Buettner, M., Prasad, R., Philipose, M., & Wetherall, D. (2009). Recognizing daily activities with RFID-based sensors. In *Proceedings of the 11th international conference on ubiquitous computing* (pp. 51–60). ACM. Retrieved 2015-01-27, from http://dl.acm.org/citation.cfm?id=1620553

*Busser, H. J., Ott, J., van Lummel, R. C., Uiterwaal, M., & Blank, R. (1997). Ambulatory monitoring of children's activity. *Medical engineering & physics*, *19*(5), 440–445.

*Bussmann, H. B., Reuvekamp, P. J., Veltink, P. H., Martens, W. L., & Stam, H. J. (1998). Validity and reliability of measurements obtained with an activity monitor in people with and without a transtibial amputation. *Physical therapy*, *78*(9), 989–998. Retrieved 2015-06-12, from http://ptjournal.apta.org/content/78/9/989.short

Bussmann, J., Veltink, P., Martens, W., & Stam, H. (1994). Activity monitoring with accelerometers. In *Dynamic analysis using body fixed sensors. congress book* (pp. 13–17).

*Bussmann, J. B., Tulen, J. H., van Herel, E. C., & Stam, H. J. (1998). Quantification of physical activities by means of ambulatory accelerometry: a validation study. *Psychophysiology*, *35*(05), 488–496.

*Bussmann, J. B. J., Van de Laar, Y. M., Neeleman, M. P., & Stam, H. J. (1998). Ambulatory accelerometry to quantify motor behaviour in patients after failed back surgery: a validation study. *Pain*, *74*(2), 153–161. Retrieved 2015-06-12, from http://www.sciencedirect.com/science/article/pii/S0304395997001619

*Bussmann, J. B. J., Veltink, P. H., Koelma, F., Van Lummel, R. C., & Stam, H. J. (1995). Ambulatory monitoring of mobility-related activities: the initial phase of the development of an activity monitor. *Eur J Phys Med Rehabil*, *5*(1), 2–7.

Buttussi, F., & Chittaro, L. (2008). MOPET: A context-aware and user-adaptive wearable system for fitness training. *Artificial Intelligence in Medicine*, *42*(2), 153–163. Retrieved 2015-01-19, from http://linkinghub.elsevier.com/retrieve/pii/S0933365707001480 doi: 10.1016/j.artmed.2007.11.004

Campbell, L., & Bobick, A. (1995). Recognition of human body motion using phase space constraints. In *Computer vision, 1995. proceedings., fifth international*

conference on (pp. 624–630). Cambridge, MA, USA. doi: 10.1109/ICCV.1995.466880

Chambers, G. S., Venkatesh, S., & West, G. A. W. (2004). Automatic labeling of sports video using umpire gesture recognition. *Computer Science, Lecture Notes in*, *1*, 859–867.

Chambers, G. S., Venkatesh, S., West, G. A. W., & Bui, H. H. (2002). Hierarchical recognition of intentional human gestures for sports video annotation. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on* (Vol. 2, pp. 1082–1085). doi: 10.1109/ICPR.2002.1048493

Chawla, N. V. (2005). Data mining for imbalanced datasets: An overview. In O. Maimon & L. Rokach (Eds.), *Data mining and knowledge discovery handbook* (pp. 853–867). Springer.

*Chen, K. Y., & Bassett, D. R. (2005). The Technology of Accelerometry-Based Activity Monitors: Current and Future:. *Medicine & Science in Sports & Exercise*, *37*(Supplement), S490–S500. Retrieved 2015-06-12, from http://content.wkhealth.com/linkback/openurl?sid=WKPTLP:landingpage&an=00005768-200511001-00002 doi: 10.1249/01.mss.0000185571.49104.82

*Chen, Y., Hall, S., McDaid, L., Buiu, O., & Kelly, P. (2008). A solid-state neuron for spiking neural network implementation. *Engineering Letters*, *16*(1), 83–89.

*Chen, Y., & Xue, Y. (2015). A deep learning approach to human activity recognition based on single accelerometer. In *Systems, man, and cybernetics (smc), 2015 ieee international conference on* (pp. 1488–1492). IEEE. doi: 10.1109/SMC.2015.263

*Chernbumroong, S., Cang, S., & Yu, H. (2015). Genetic algorithm-based classifiers fusion for multisensor activity recognition of elderly people. *IEEE Journal of Biomedical and Health Informatics*, *19*(1), 282–289. doi: 10.1109/JBHI.2014.2313473

*Cho, Y., Nam, Y., Choi, Y., & Cho, W. (2008). SmartBuckle: human activity recognition using a 3-axis accelerometer and a wearable camera. In *Proceedings of the 2nd International Workshop on Systems and Networking Support for Health Care and Assisted Living Environments* (p. 7). New York, NY: ACM. Retrieved 2015-08-10, from http://dl.acm.org/citation.cfm?id=1515747

*Choudhury, T., Consolvo, S., Harrison, B., Hightower, J., LaMarca, A., LeGrand, L., ... others (2008). The mobile sensing platform: An embedded activity recognition system. *Pervasive Computing, IEEE*, *7*(2), 32–41.

*Chuang, F.-C., Wang, J.-S., Yang, Y.-T., & Kao, T.-P. (2012). A wearable activity sensor system and its physical activity classification scheme. In *The 2012 international joint conference on neural networks (IJCNN)* (pp. 1–6). doi: 10.1109/IJCNN.2012.6252581

*Coley, B., Najafi, B., Paraschiv-Ionescu, A., & Aminian, K. (2005). Stair climbing detection during daily physical activity using a miniature gyroscope. *Gait & Posture*, *22*(4), 287–294. Retrieved 2015-06-12, from http://linkinghub.elsevier.com/retrieve/pii/S0966636204002553 doi: 10.1016/j.gaitpost.2004.08.008

Companjen, B. (2009). Classification methods for activity recognition. In *11th Twente Student Conference on IT*. Retrieved 2012-10-21, from http://referaat.cs.utwente.nl/TSConIT/download.php?id=523

Connaghan, D., Kelly, P., O'Connor, N. E., Gaffney, M., Walsh, M., & O'Mathuna, C. (2011). Multi-sensor classification of tennis strokes. In *Sensors, 2011 IEEE* (pp. 1437–1440). IEEE. Retrieved 2014-06-03, from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6127084

*Consolvo, S., McDonald, D. W., Toscos, T., Chen, M. Y., Froehlich, J., Harrison, B., ... others (2008). Activity sensing in the wild: a field trial of ubifit garden. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1797–1806). ACM. Retrieved 2015-06-04, from http://dl.acm.org/citation.cfm?id=1357335

Cook, D. J., Augusto, J. C., & Jakkula, V. R. (2009). Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing*, *5*(4), 277–298. Retrieved 2012-09-07, from http://www.sciencedirect.com/science/article/pii/S157411920900025X doi: 10.1016/j.pmcj.2009.04.001

*Coskun, D., Incel, O. D., & Ozgovde, A. (2015). Phone position/placement detection using accelerometer: Impact on activity recognition. In *Intelligent sensors, sensor networks and information processing (ISSNIP), 2015 IEEE tenth international conference on* (pp. 1–6). IEEE.

*Culhane, K. M., Lyons, G. M., Hilton, D., Grace, P. A., & Lyons, D. (2004). Long-term mobility monitoring of older adults using accelerometers in a clinical environment. *Clinical Rehabilitation*, *18*(3), 335–343. Retrieved 2015-06-12, from http://cre.sagepub.com/cgi/doi/10.1191/0269215504cr734oa doi: 10.1191/0269215504cr734oa

*Dalton, A., & OLaighin, G. (2013). Comparing supervised learning techniques on the task of physical activity recognition. *IEEE Journal of Biomedical and Health Informatics*, *17*(1), 46–52. doi: 10.1109/TITB.2012.2223823

Darell, T., & Pentland, A. (1996). Active gesture recognition using partially observable markov decision processes. In *Pattern recognition, 1996., proceedings of the 13th international conference on* (Vol. 3, pp. 984–988). IEEE. Retrieved 2015-01-27, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.47.6423&rep=rep1&type=pdf

Davey, N., Anderson, M., & James, D. A. (2008). Validation trial of an accelerometer-based sensor platform for swimming. *Sports Technology*, *1*(4), 202–207. doi: 10.1002/jst.59

de Garis, H., Nawa, N., Hough, M., & Korkin, M. (1999). Evolving an optimal de/-convolution function for the neural net modules of atr's artificial brain project. In *Neural networks, 1999. ijcnn '99. international joint conference on* (Vol. 1, p. 438-443). doi: 10.1109/IJCNN.1999.831535

*Degen, T., Jaeckel, H., Rufer, M., & Wyss, S. (2003). SPEEDY: A Fall Detector in a Wrist Watch. In *ISWC* (pp. 184–189). Retrieved 2015-06-13, from http://www.imes.hsr.ch/fileadmin/user_upload/imes.hsr.ch/Publikationen/10_ISWC03.pdf

De la Torre, F., Hodkins, J., Montano, J., Valcarcel, S., Forcada, R., & Macey, J.

(2009). *Quality of life grand challenge kitchen capture.* WWW. Retrieved from http://kitchen.cs.cmu.edu/main.php

*de la Vega, L. G. M., Raghuraman, S., Balasubramanian, A., & Prabhakaran, B. (2013). Exploring unconstrained mobile sensor based human activity recognition. In *3rd international workshop on mobile sensing* (pp. 8–11).

*Deng, W., Zheng, Q., & Wang, Z. (2014). Cross-person activity recognition using reduced kernel extreme learning machine. *Neural Networks*, *53*, 1–7. doi: 10.1016/j.neunet.2014.01.008

*Derawi, M., & Bours, P. (2013). Gait and activity recognition using commercial phones. *Computers & Security*, *39*, 137–144. doi: 10.1016/j.cose.2013.07.004

Devert, A., Bredeche, N., & Schoenauer, M. (2007). Unsupervised learning of echo state networks: A case study in artificial embryogeny. In *International conference on artificial evolution (evolution artificielle)* (pp. 278–290). Springer.

*Doukas, C., & Maglogiannis, I. (2008). Advanced patient or elder fall detection based on movement and sound data. In *Pervasive Computing Technologies for Healthcare, 2008. PervasiveHealth 2008. Second International Conference on* (pp. 103–107). IEEE. Retrieved 2015-06-13, from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4571042

*Doukas, C., & Maglogiannis, I. (2011). Emergency Fall Incidents Detection in Assisted Living Environments Utilizing Motion, Sound, and Visual Perceptual Components. *IEEE Transactions on Information Technology in Biomedicine*, *15*(2), 277–289. Retrieved 2015-06-13, from http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5623343 doi: 10.1109/TITB.2010.2091140

Drummond, C., Holte, R. C., & others. (2003). C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In *Workshop on learning from imbalanced datasets II* (Vol. 11). Citeseer.

*Dunnewold, R., Jacobi, C., & Van Hilten, J. (1997). Quantitative assessment of bradykinesia in patients with Parkinson's disease. *Journal of neuroscience methods*, *74*(1), 107–112.

*Ermes, M., Parkka, J., & Cluitmans, L. (2008). Advancing from offline to online activity recognition with wearable sensors. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE* (pp. 4451–4454). IEEE.

*Ermes, M., Parkka, J., Mantyjarvi, J., & Korhonen, I. (2008). Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. *IEEE Transactions on Information Technology in Biomedicine*, *12*(1), 20–26. doi: 10.1109/TITB.2007.899496

*Fahrenberg, J. (1996). Computer assisted Psychological and Psychophysiological Methods in Monitoring and Field Studies. *Ambulatory assessment: Issues and perspectives*.

*Fahrenberg, J., Foerster, F., Smeja, M., & Muller, W. (1997). Assessment of posture and motion by multichannel piezoresistive accelerometer recordings. *Psychophysiology*, *34*(5), 607–612.

Fels, S., & Hinton, G. (1993). Glove-talk: a neural network interface between a data-glove and a speech synthesizer. *IEEE Transactions on Neural Networks*,

4(1), 2–8. doi: 10.1109/72.182690

Fette, G., & Eggert, J. (2005). Short term memory and pattern matching with simple echo state networks. In *Artificial neural networks: Biological inspirationsâĂŞICANN 2005* (pp. 13–18). Springer. Retrieved 2015-01-16, from http://link.springer.com/chapter/10.1007/11550822_3

*Fida, B., Bernabucci, I., Bibbo, D., Conforto, S., & Schmid, M. (2015). Varying behavior of different window sizes on the classification of static and dynamic physical activities from a single accelerometer. *Medical Engineering & Physics*, *37*(7), 705–711. doi: 10.1016/j.medengphy.2015.04.005

*Figo, D., Diniz, P., Ferreira, D. R., & Cardoso, J. M. P. (2010). Preprocessing techniques for context recognition from accelerometer data. *Personal & Ubiquitous Computing*, *14*(7), 645–662. doi: 10.1007/s00779-010-0293-9

Flexer, A. (1996). Statistical evaluation of neural network experiments: Minimum requirements and current practice. *Cybernetics and Systems Research*, *1*, 1005–1008.

*Foerster, F., & Fahrenberg, J. (2000). Motion pattern and posture: correctly assessed by calibrated accelerometers. *Behavior research methods, instruments and computers*, *32*(3), 450–457.

*Foerster, F., Smeja, M., & Fahrenberg, J. (1999). Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring. *Human Behavior, Computers in*, *15*(5), 571–583. doi: 10.1016/S0747-5632(99)00037-0

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *The Journal of machine learning research*, *3*, 1289–1305.

Foxlin, E., Harrington, M., & Pfeifer, G. (1998). Constellation: A wide-range wireless motion-tracking system for augmented reality and virtual set applications. In *Proceedings of the 25th annual conference on computer graphics and interactive techniques* (pp. 371–378).

*Frank, K., Rockl, M., Nadales, M. J. V., Robertson, P., & Pfeifer, T. (2010). Comparison of exact static and dynamic bayesian context inference methods for activity recognition. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on* (pp. 189–195). IEEE.

*Fujimoto, T., Nakajima, H., Tsuchiya, N., Marukawa, H., Kuramoto, K., Kobashi, S., & Hata, Y. (2013). Wearable human activity recognition by electrocardiograph and accelerometer. In *Multiple-valued logic (ismvl), 2013 ieee 43rd international symposium on* (pp. 12–17). IEEE. doi: 10.1109/ISMVL.2013.60

Gallicchio, C., & Micheli, A. (2016). A Reservoir Computing Approach for Human Gesture Recognition from Kinect Data. In *AI* AAL@ AI* IA* (pp. 33–42).

Galtier, M. (2014). Ideomotor feedback control in a recurrent neural network. *arXiv preprint arXiv:1402.3563*, *Preprint*, 1.

*Ganti, R. K., Jayachandran, P., Abdelzaher, T. F., & Stankovic, J. A. (2006). Satire: a software architecture for smart attire. In *Proceedings of the 4th international conference on Mobile systems, applications and services* (pp. 110–123). ACM. Retrieved 2015-08-10, from http://dl.acm.org/citation.cfm?id=1134693

*Gao, L., Bourke, A. K., & Nelson, J. (2014). Evaluation of accelerometer based

multi-sensor versus single-sensor activity recognition systems. *Medical Engineering & Physics*, *36*(6), 779–785. doi: 10.1016/j.medengphy.2014.02.012

Gerstner, W., & Kistler, W. M. (2002). *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge, MA: Cambridge University Press.

Ghasemzadeh, H., Loseu, V., & Jafari, R. (2009). Wearable coach for sport training: A quantitative model to evaluate wrist-rotation in golf. *Journal of Ambient Intelligence and Smart Environments*, *1*(2), 173–184. Retrieved 2015-01-19, from http://iospress.metapress.com/index/B5503511V8912N18.pdf

Godfrey, A., Bourke, A., Olaighin, G., van de Ven, P., & Nelson, J. (2011). Activity classification using a single chest mounted tri-axial accelerometer. *Medical Engineering & Physics*, *33*(9), 1127–1135. doi: 10.1016/j.medengphy.2011.05.002

Godfrey, A., Conway, R., Meagher, D., & ÓLaighin, G. (2008). Direct measurement of human movement by accelerometry. *Medical Engineering & Physics*, *30*(10), 1364–1386. doi: 10.1016/j.medengphy.2008.09.005

Goodman, E., & Ventura, D. (2006). Spatiotemporal pattern recognition via liquid state machines. In *International joint conference on neural networks* (p. 3848-3853). Vancouver, BC. doi: 10.1109/IJCNN.2006.246880

*Guo, H., Chen, L., Shen, Y., & Chen, G. (2014). Activity recognition exploiting classifier level fusion of acceleration and physiological signals. In *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing: Adjunct publication* (pp. 63–66). ACM. doi: 10.1145/2638728.2638777

*Gupta, P., & Dallas, T. (2014). Feature selection and activity recognition system using a single triaxial accelerometer. *IEEE Transactions on Biomedical Engineering*, *61*(6), 1780–1786. doi: 10.1109/TBME.2014.2307069

*Gyorbiro, N., Fabian, A., & Homanyi, G. (2009). An Activity Recognition System For Mobile Phones. *Mobile Networks and Applications*, *14*(1), 82–91. Retrieved 2015-08-10, from http://link.springer.com/10.1007/s11036-008-0112-y doi: 10.1007/s11036-008-0112-y

*Hanai, Y., Nishimura, J., & Kuroda, T. (2009). Haar-like filtering for human activity recognition using 3d accelerometer. In *Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop, 2009. DSP/SPE 2009. IEEE 13th* (pp. 675–678). IEEE.

*Harding, J. W., Mackintosh, C. G., Hahn, A. G., & James, D. A. (2008). Classification of Aerial Acrobatics in Elite Half-Pipe Snowboarding Using Body Mounted Inertial Sensors (P237). In *The Engineering of Sport 7* (pp. 447–456). Springer.

Harms, H., Amft, O., TrÃűster, G., & Roggen, D. (2008). SMASH: a distributed sensing and processing garment for the classification of upper body postures. In *Proceedings of the ICST 3rd international conference on body area networks* (pp. 22:1–22:8). ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). Retrieved 2014-06-03, from http://dl.acm.org/citation.cfm?id=1460257.1460287

*He, Z., & Jin, L. (2009). Activity recognition from acceleration data based on

discrete consine transform and svm. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on* (pp. 5041–5044). IEEE.

*He, Z., Liu, Z., Jin, L., Zhen, L.-X., & Huang, J.-C. (2008). Weightlessness feature – a novel feature for single tri-axial accelerometer based activity recognition. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on* (pp. 1–4). IEEE.

*He, Z.-Y., & Jin, L.-W. (2008). Activity recognition from acceleration data using AR model representation and SVM. In *Machine Learning and Cybernetics, 2008 International Conference on* (Vol. 4, pp. 2245–2250). IEEE.

Heinz, E. A., Kunze, K., Gruber, M., Bannach, D., & Lukowicz, P. (2003). Using Wearable Sensors for Real-time Recognition Tasks in Games of Martial Arts An Initial Experiment. In *Proceedings of the 2006 ieee symposium on computational intelligence and games*. Veldhoven, The Netherlands: Springer-Verlag.

*Heinz, E. A., Kunze, K. S., Sulistyo, S., Junker, H., Lukowicz, P., & Tröster, G. (2003). Experimental evaluation of variations in primary features used for accelerometric context recognition. In *Ambient Intelligence* (pp. 252–263). Springer.

Henery, R. J. (1994). Methods for comparison. In D. Michie, D. J. Spiegelhalter, & C. C. Taylor (Eds.), *Machine learning, neural and statistical classification*. New York, NY, USA: Ellis Horwood.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *Management Information Systems Quarterly*, *28*(1), 75–106.

*Hoettinger, H., Mally, F., & Sabo, A. (2016). Activity recognition in surfing - a comparative study between hidden markov model and support vector machine. *Procedia Engineering*, *147*, 912–917. doi: 10.1016/j.proeng.2016.06.279

*Hu, L., Chen, Y., Wang, S., & Chen, Z. (2014). b-COELM: A fast, lightweight and accurate activity recognition model for mini-wearable devices. *Pervasive and Mobile Computing*, *15*, 200–214. doi: 10.1016/j.pmcj.2014.06.002

Hunt, D. P. L. (2009). *A heuristic method to distinguish horse rider mounts using a single wrist mounted inertial sensor* (Unpublished master's thesis). Auckland University of Technology, Auckland, New Zealand.

Hunt, D. P. L., & Parry, D. (2015). Using Echo State Networks to Classify Unscripted, Real-World Punctual Activity. In *Engineering Applications of Neural Networks* (pp. 369–378). Springer.

Hunt, D. P. L., & Parry, D. (2016). The role of classification in the development of wearable coaching devices. In *The 13th australasian conference on mathematics and computers in sport* (Vol. 1, pp. 45–49). ANZIAM MathSport. Retrieved from http://www.anziam.org.au/tiki-download_file.php?fileId=86

Hunt, D. P. L., Parry, D., & Schliebs, S. (2014). Exploring the applicability of reservoir methods for classifying punctual sports activities using on-body sensors. In *ACSC 2014*. Auckland, NZ: Australian Computer Society.

*Huynh, T., Blanke, U., & Schiele, B. (2007). Scalable recognition of daily activities with wearable sensors. In *Location-and context-awareness* (pp. 50–67). Springer.

*Huynh, T., & Schiele, B. (2005). Analyzing features for activity recognition. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies* (pp. 159–163). ACM.

*Huynh, T., & Schiele, B. (2006a). Towards less supervision in activity recognition from wearable sensors. In *Wearable Computers, 2006 10th IEEE International Symposium on* (pp. 3–10). IEEE.

*Huynh, T., & Schiele, B. (2006b). Unsupervised discovery of structure in activity data using multiple eigenspaces. In *Location-and Context-Awareness* (pp. 151–167). Springer.

*Hwang, J. Y., Kang, J. M., Jang, Y. W., & Kim, H. C. (2004). Development of novel algorithm and real-time monitoring ambulatory system using Bluetooth module for fall detection in the elderly. In *Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE* (Vol. 1, pp. 2204–2207). IEEE.

*Iglesias, J., Cano, J., Bernados, A. M., & Casa, J. R. (2011). A ubiquitous activity-monitor to prevent sedentariness. In *Proceedings of the 2011 ieee international conference on pervasive computing and communications* (pp. 319–321). Seattle, WA: IEEE. doi: 10.1109/PERCOMW.2011.5766894

*Ignatov, A. D., & Strijov, V. V. (2016). Human activity recognition using quasiperiodic time series collected from a single tri-axial accelerometer. *Multimedia Tools and Applications*, *75*(12), 7257–7270. doi: 10.1007/s11042-015-2643-0

Ilies, I., Jaeger, H., Kosuchinas, O., Rincon, M., Sakenas, V., & Vaskevicius, N. (2007). *Stepping forward through echoes of the past: forecasting with Echo State Networks.* Retrieved from http://www.neural-forecasting-competition.com/downloads/methods/27-NN3_Herbert_Jaeger_report.pdf

Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks–with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Reports*, *148*, 34.

Jaeger, H. (2003). Adaptive nonlinear system identification with echo state networks. *networks*, *8*, 9. Retrieved 2014-06-09, from https://papers.nips.cc/paper/2318-adaptive-nonlinear-system-identification-with-echo-state-networks.pdf

Jaeger, H. (2005). *A tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach*. German National Research Center for Information Technology. Retrieved from http://www.pdx.edu/sites/www.pdx.edu.sysc/files/Jaeger_TrainingRNNsTutorial.2005.pdf

Jaeger, H., & Haas, H. (2004). Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, *304*(5667), 78–80. doi: 10.1126/science.1091277

Jaeger, H., Lukosevicius, M., Popovici, D., & Siewert, U. (2007). Optimization and applications of echo state networks with leaky- integrator neurons. *Neural Networks*, *20*(3), 335–352. doi: 10.1016/j.neunet.2007.04.016

Jarvinen, P. H. (2000). Research questions guiding selection of an appropriate research method. In *Ecis 2000 proceedings* (p. 26). Retrieved 2015-12-18, from

http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1023&context=ecis2000

*Jatoba, L. C., Grossmann, U., Kunze, C., Ottenbacher, J., & Stork, W. (2008). Context-aware mobile health monitoring: Evaluation of different pattern recognition methods for classification of physical activity. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE* (pp. 5250–5253). IEEE.

*Jia, R., & Liu, B. (2013). Human daily activity recognition by fusing accelerometer and multi-lead ECG data. In *Signal processing, communication and computing (ICSPCC), 2013 IEEE international conference on* (pp. 1–4). IEEE.

*Jiang, W., & Yin, Z. (2015). Human activity recognition using wearable sensors by deep convolutional neural networks. In *Proceedings of the 23rd acm international conference on multimedia* (pp. 1307–1310). ACM Press. doi: 10.1145/2733373.2806333

*Jin, G. H., Lee, S. B., & Lee, T. S. (2008). Context Awareness of Human Motion States Using Accelerometer. *Journal of Medical Systems*, *32*(2), 93–100. Retrieved 2015-08-10, from http://link.springer.com/10.1007/s10916-007-9111-y doi: 10.1007/s10916-007-9111-y

*Junker, H., Amft, O., Lukowicz, P., & Troster, G. (2008). Gesture spotting with body-worn inertial sensors to detect user activities. *Pattern Recognition*, *41*(6), 2010–2024. doi: 10.1016/j.patcog.2007.11.016

Junker, H., Lukowicz, P., & Troster, G. (2004). Continuous recognition of arm activities with body-worn inertial sensors. In *Wearable computers, 2004 8th IEEE international symposium on* (Vol. 1, pp. 188–189).

*Kamada, M., Shiroma, E. J., Harris, T. B., & Lee, I. (2016). Comparison of physical activity assessed using hip- and wrist-worn accelerometers. *Gait & Posture*, *44*, 23–28. doi: 10.1016/j.gaitpost.2015.11.005

*Kao, T.-P., Lin, C.-W., & Wang, J.-S. (2009). Development of a portable activity detector for daily activity recognition. In *Industrial Electronics, 2009. ISIE 2009. IEEE International Symposium on* (pp. 115–120). IEEE.

*Karantonis, D., Narayanan, M., Mathie, M., Lovell, N., & Celler, B. (2006). Implementation of a Real-Time Human Movement Classifier Using a Triaxial Accelerometer for Ambulatory Monitoring. *IEEE Transactions on Information Technology in Biomedicine*, *10*(1), 156–167. doi: 10.1109/TITB.2005.856864

Kawahara, Y., Sugimoto, C., Arimitsu, S., Itao, T., Morandini, A., Morikawa, H., & Aoyama, T. (2005). Context inference techniques for a wearable exercise support system. In *ACM SIGGRAPH 2005 posters* (p. 91). ACM Press. Retrieved 2015-01-19, from http://portal.acm.org/citation.cfm?doid=1186954.1187058 doi: 10.1145/1186954.1187058

*Kern, N., Schiele, B., & Schmidt, A. (2003). Multi-sensor Activity Context Detection for Wearable Computing. In E. Aarts, R. Collier, E. van Loenen, & B. de Ruyter (Eds.), *Ambient Intelligence* (Vol. 2875, pp. 220–232). Springer.

*Khan, A. M., Lee, Y. K., & Lee, S. Y. (2010). Accelerometer's position free human activity recognition using a hierarchical recognition model. In *e-Health Networking Applications and Services (Healthcom), 2010 12th IEEE International Conference on* (pp. 296–301). IEEE.

*Khan, A. M., Young-Koo, L., Lee, S. Y., & Tae-Seong, K. (2010). A Triaxial Accelerometer-Based Physical-Activity Recognition via Augmented-Signal Features and a Hierarchical Recognizer. *IEEE Transactions on Information Technology in Biomedicine*, *14*(5), 1166–1172. doi: 10.1109/TITB.2010.2051955

*Kiani, K., Snijders, C., & Gelsema, E. (1997). Computerized analysis of daily life motor activity for ambulatory monitoring. *Technology and health care: official journal of the European Society for Engineering and Medicine*, *5*(4), 307–318.

*Kiani, K., Snijders, C. J., & Gelsema, E. S. (1998). Recognition of daily life motor activity classes using an artificial neural network. *Archives of physical medicine and rehabilitation*, *79*(2), 147–154.

*Kim, T., Cho, J., & Kim, J. T. (2013). Mobile motion sensor-based human activity recognition and energy expenditure estimation in building environments. In A. Hakansson, M. Höjer, R. J. Howlett, & L. C. Jain (Eds.), *Sustainability in energy and buildings: Proceedings of the 4th international conference in sustainability in energy and buildings (SEBt'12)* (Vol. 22, pp. 987–993). Springer Berlin Heidelberg. (DOI: 10.1007/978-3-642-36645-1_87)

Kim, Y., & Ling, H. (2009). Human activity classification based on micro-doppler signatures using a support vector machine. *IEEE Transactions on Geoscience and Remote Sensing*, *47*(5), 1328–1337. doi: 10.1109/TGRS.2009.2012849

Kjeldskov, J., & Graham, C. (2003). A review of mobile HCI research methods. *Computer Science, Lecture Notes in*, *1*, 317–335.

Koskimaki, H., Huikari, V., Siirtola, P., Laurinen, P., & Roning, J. (2009). Activity recognition using a wrist-worn inertial measurement unit: A case study for industrial assembly lines. In *Control and automation, 2009. MED'09. 17th mediterranean conference on* (pp. 401–405). IEEE. Retrieved 2015-01-27, from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5164574

Koutnik, J., Greff, K., Gomez, F., & Schmidhuber, J. (2014). A clockwork RNN. *arXiv preprint arXiv:1402.3511*, *Preprint*, 1.

*Krause, A., Ihmig, M., Rankin, E., Leong, D., Gupta, S., Siewiorek, D., ... Sengupta, U. (2005). Trading off Prediction Accuracy and Power Consumption for Context-Aware Wearable Computing. In *Proceedings of the ninth ieee international symposium on wearable computers* (pp. 20–26). IEEE. doi: 10.1109/ISWC.2005.52

*Krause, A., Siewiorek, D. P., Smailagic, A., & Farringdon, J. (2003). Unsupervised, dynamic identification of physiological and activity context in wearable computing. In *null* (p. 88). IEEE.

*Kunze, K., & Lukowicz, P. (2008). Dealing with sensor displacement in motion-based onbody activity recognition systems. In *Ubiquitous Computing, International conference on - 10th (SESSION: Activity Sensing)* (Vol. 344 archive, pp. 20–29). Seoul, Korea: ACM New York, NY, USA. doi: 10.1145/1409635.1409639

*Kunze, K., & Lukowicz, P. (2014). Sensor placement variations in wearable activity recognition. *IEEE Pervasive Computing*, *13*(4), 32–41.

*Kwapisz, J. R., Weiss, G. M., & Moore, S. A. (2011). Activity recogni-

tion using cell phone accelerometers. *SIGKDD Explor. Newsl.*, *12*(2), 74–82. Retrieved 2012-08-21, from http://doi.acm.org.ezproxy.aut.ac.nz/10.1145/1964897.1964918 doi: 10.1145/1964897.1964918

*Kwon, Y., Kang, K., & Bae, C. (2015). Analysis and evaluation of smartphone-based human activity recognition using a neural network approach. In *Neural networks (IJCNN), 2015 international joint conference on* (pp. 1–5). IEEE.

Lang, P., Kusej, A., Pinz, A., & Brasseur, G. (2002). Inertial tracking for mobile augmented reality. In *Instrumentation and measurement technology conference, 2002. imtc/2002. proceedings of the 19th ieee* (Vol. 2, pp. 1583–1587).

Lantz, B. (2013). *Machine learning with R*. Packt Publishing Ltd.

Lara, O. D., & Labrador, M. A. (2013). A Survey on Human Activity Recognition using Wearable Sensors. *IEEE Communications Surveys & Tutorials*, *15*(3), 1192–1209. Retrieved 2015-06-02, from http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6365160 doi: 10.1109/SURV.2012.110112.00192

*Lara, O. D., Perez, A. J., Labrador, M. A., & Posada, J. D. (2012). Centinela: A human activity recognition system based on acceleration and vital sign data. *Pervasive and Mobile Computing*, *8*(5), 717–729. doi: 10.1016/j.pmcj.2011.06.004

Larose, D. T. (2014). *Discovering knowledge in data: an introduction to data mining*. John Wiley & Sons.

Lau, H., & Tong, K. (2008). The reliability of using accelerometer and gyroscope for gait event identification on persons with dropped foot. *Gait & Posture*, *27*(2), 248–257. doi: 10.1016/j.gaitpost.2007.03.018

*Laudanski, A., Brouwer, B., & Li, Q. (2015). Activity classification in persons with stroke based on frequency features. *Medical Engineering & Physics*, *37*(2), 180–186. doi: 10.1016/j.medengphy.2014.11.008

Lee, H., & Kim, J. (1998). Gesture spotting from continuous hand motion. *Pattern Recognition Letters*, *19*(5-6), 513–520.

Lee, J., & Ha, I. (2001). Real-time motion capture for a human body using accelerometers. *Robotica*, *19*(06), 601–610. doi: 10.1017/S0263574701003319

*Lee, S., & Mase, K. (2002). Activity and location recognition using wearable sensors. *IEEE Pervasive Computing*, *1*(3), 24–32. doi: 10.1109/MPRV.2002.1037719

*Lee, S. H., Park, H. D., Hong, S. Y., Lee, K. J., & Kim, Y. H. (2003). A study on the activity classification using a triaxial accelerometer. In *Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE* (Vol. 3, pp. 2941–2943). IEEE.

*Lee, Y.-S., & Cho, S.-B. (2011). Activity Recognition Using Hierarchical Hidden Markov Models on a Smartphone with 3d Accelerometer. In E. Corchado, M. KurzyÅĎski, & M. WoÅžniak (Eds.), *Hybrid Artificial Intelligent Systems* (pp. 460–467). Springer Berlin Heidelberg.

*Leonard, M., Godfrey, A., Silberhorn, M., Conroy, M., Donnelly, S., Meagher, D., & Olaighin, G. (2007). Motion Analysis in Delirium: A Novel Method of Clarifying Motoric Subtypes. *Neurocase*, *13*(4), 272–277. doi: 10.1080/

13554790701646233

*Lester, J., Choudhury, T., & Borriello, G. (2006). A practical approach to recognizing physical activities. In *Pervasive Computing* (pp. 1–16). Springer. Retrieved 2015-09-08, from http://link.springer.com/chapter/10.1007/11748625_1

Lester, J., Choudhury, T., Kern, N., Borriello, G., & Hannaford, B. (2005). A hybrid discriminative/generative approach for modeling human activities. In *IJCAI* (Vol. 5, pp. 766–772). doi: 10.1.1.77.5776

*Leutheuser, H., Schuldhaus, D., & Eskofier, B. M. (2013). Hierarchical, multi-sensor based classification of daily life activities: Comparison with state-of-the-art algorithms using a benchmark dataset. *PLoS ONE*, *8*(10), e75196. doi: 10.1371/journal.pone.0075196

*Li, C., Chen, Y., Chen, W., Huang, P., & Chu, H. (2013). Sensor-embedded teeth for oral activity recognition. In *Proceedings of the 2013 international symposium on wearable computers* (pp. 41–44). ACM.

Lichman, M. (2013). *UCI machine learning repository*. University of California, Irvine, School of Information and Computer Sciences. Retrieved from http://archive.ics.uci.edu/ml

Lin, S., Ying, K., Chen, S., & Lee, Z. (2008). Particle swarm optimization for parameter determination and feature selection of support vector machines. *Expert Systems with Applications*, *35*(4), 1817–1824. doi: 10.1016/j.eswa.2007.08.088

*Lindemann, U., Hock, A., Stuber, M., Keck, W., & Becker, C. (2005). Evaluation of a fall detector based on accelerometers: A pilot study. *Medical and Biological Engineering and Computing*, *43*(5), 548–551.

Liu, D., He, C., Zhao, Q., Yang, Z., Hao, Y., & Yan, G. (2014). Digital signal processing for a micromachined vibratory gyroscope based on a three dimensional adaptive filter demodulator. *Measurement*, *50*, 198–202. doi: 10.1016/j.measurement.2013.12.025

Liu, X., Wu, J., & Zhou, Z. (2009). Exploratory Undersampling for Class-Imbalance Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *39*(2), 539–550. doi: 10.1109/TSMCB.2008.2007853

*Lockhart, J. W., & Weiss, G. M. (2014). The benefits of personalized smartphone-based activity recognition models. In M. Zaki, Z. Obradovic, P. N. Tan, A. Banerjee, C. Kamath, & S. Parthasarathy (Eds.), *Proceedings of the 2014 SIAM international conference on data mining* (pp. 614–622). Society for Industrial and Applied Mathematics. (DOI: 10.1137/1.9781611973440.71)

*Logan, B., Healey, J., Philipose, M., Tapia, E. M., & Intille, S. (2007). A Long-Term Evaluation of Sensing Modalities for Activity Recognition. In J. Krumm, G. D. Abowd, A. Seneviratne, & T. Strang (Eds.), *UbiComp 2007: Ubiquitous Computing* (pp. 483–500). Springer Berlin Heidelberg.

Lombriser, C., Bharatula, N. B., Roggen, D., & Troster, G. (2007). On-body activity recognition in a dynamic sensor network. In *Proceedings of the ICST 2nd international conference on body area networks* (p. 17). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). Retrieved 2015-01-23, from http://dl.acm.org/citation.cfm?id=1460249

*Long, X., Yin, B., & Aarts, R. M. (2009). Single-accelerometer-based daily physical activity classification. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE* (pp. 6107–6110). IEEE.

*Lubina, P., & Rudzki, M. (2015). Artificial neural networks in accelerometer-based human activity recognition. In *Mixed design of integrated circuits & systems (MIXDES), 2015 22nd international conference* (pp. 63–68). IEEE.

Lukoševičius, M. (2012). A practical guide to applying echo state networks. In *Neural networks: Tricks of the trade* (pp. 659–686). Bremen, Germany: Springer.

Lukoševičius, M., & Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, *3*(3), 127–149. doi: 10.1016/j.cosrev.2009.03.005

*Lukowicz, P., Ward, J., Junker, H., Stäger, M., Tröster, G., Atrash, A., & Starner, T. (2004). Recognizing workshop activity using body worn microphones and accelerometers. In *Pervasive computing* (pp. 18–32). Springer.

Maass, W. (2010). Liquid state machines: motivation, theory, and applications. In S. B. Cooper & A. Sorbi (Eds.), *Computability in Context: Computation and Logic in the Real World* (pp. 275–296). World Scientific.

Maass, W., Natschläger, T., & Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, *14*(11), 2531–2560. doi: 10.1162/089976602760407955

Madgwick, S. O. (2010). An efficient orientation filter for inertial and inertial/magnetic sensor arrays. *Report x-io and University of Bristol (UK)*, *1*, 1.

Maji, K. B., Sree, U. S., Kar, R., Mandal, D., & Ghoshal, S. P. (2015). Butterworth filter design using seeker optimization algorithm. In *Science and Technology (TICST), 2015 International Conference on* (pp. 298–302). IEEE.

*Mandal, I., Happy, S. L., Behera, D. P., & Routray, A. (2014). A framework for human activity recognition based on accelerometer data. In *Confluence the next generation information technology summit (confluence), 2014 5th international conference* (pp. 600–603). IEEE.

*Mannini, A., & Sabatini, A. M. (2010). Machine Learning Methods for Classifying Human Physical Activity from On-Body Accelerometers. *Sensors*, *10*(2), 1154–1175. doi: 10.3390/s100201154

Mantyjarvi, J., Himberg, J., & Seppanen, T. (2001). Recognizing human motion with multiple acceleration sensors. In *Systems, man, and cybernetics, 2001 IEEE international conference on* (Vol. 2, pp. 747–752). doi: 10.1109/ICSMC.2001.973004

*Mantyjarvi, J., Kela, J., Korpipaa, P., & Kallio, S. (2004). Enabling fast and effortless customisation in accelerometer based gesture interaction. In *Mobile and ubiquitous multimedia, International conference on - 3rd.* College Park, Maryland: ACM. doi: 10.1145/1052380.1052385

Marceau, S. (2014). *Brave new wearable world: Crowdsourcing health, and the coming battle of bio-signals* [Blog]. Retrieved 2014-08-26, from http://venturebeat.com/2014/04/16/brave-new-wearable-world

-crowdsourcing-health-and-the-coming-battle-of-bio-signals/

*Margarito, J., Helaoui, R., Bianchi, A., Sartor, F., & Bonomi, A. (2015). User-independent recognition of sports activities from a single wrist-worn accelerometer: A template matching based approach. *IEEE Transactions on Biomedical Engineering*, 1–1. doi: 10.1109/TBME.2015.2471094

*Marin-Perianu, M., Lombriser, C., Amft, O., Havinga, P., & Traster, G. (2008). Distributed activity recognition with fuzzy-enabled wireless sensor networks. *Distributed Computing in Sensor Systems*, *1*, 296–313. Retrieved 2012-10-22, from http://www.springerlink.com/index/ut1u651151372710.pdf

Markram, H., Wang, Y., & Tsodyks, M. (1998). Differential signaling via the same axon of neocortical pyramidal neurons. *Proceedings of the National Academy of Sciences*, *95*(9), 5323–5328.

*Martín, H., Bernardos, A. M., Iglesias, J., & Casar, J. R. (2013). Activity logging using lightweight classification techniques in mobile devices. *Personal and Ubiquitous Computing*, *17*(4), 675–695. doi: 10.1007/s00779-012-0515-4

*Mathie, M. J., Celler, B. G., Lovell, D. N. H., & Coster, A. C. F. (2004). Classification of basic daily movements using a triaxial accelerometer. *Medical and Biological Engineering and Computing*, *42*(5), 679–687. doi: 10.1007/BF02347551

*Maurer, U., Smailagic, A., Siewiorek, D., & Deisher, M. (2006). Activity recognition and monitoring using multiple sensors on different body positions. In *International Workshop on Wearable and Implantable Body Sensor Networks, 2006. BSN 2006* (pp. 4–116). doi: 10.1109/BSN.2006.6

Mayagoitia, R. E., Nene, A. V., & Veltink, P. H. (2002). Accelerometer and rate gyroscope measurement of kinematics: an inexpensive alternative to optical motion analysis systems. *Biomech, J*, *35*(4), 537–42. doi: 10.1016/S0021-9290(01)00231-7

*McGlynn, D., & Madden, M. G. (2011). An Ensemble Dynamic Time Warping Classifier with Application to Activity Recognition. In M. Bramer, M. Petridis, & A. Hopgood (Eds.), *Research and Development in Intelligent Systems XXVII* (pp. 339–352). London: Springer London.

Medrano, C., Igual, R., Plaza, I., & Castro, M. (2014). Detecting falls as novelties in acceleration patterns acquired with smartphones. *PLoS ONE*, *9*(4), e94811. Retrieved from http://dx.plos.org/10.1371/journal.pone.0094811 doi: 10.1371/journal.pone.0094811

*Michahelles, F., & Schiele, B. (2005). Sensing and Monitoring Professional Skiers. *Pervasive Computing and Communications, 2005. Third Annual IEEE International Conference on*, *4*(3), 40–46.

Mici, L., Hinaut, X., & Wermter, S. (2016). Activity recognition with echo state networks using 3d body joints and objects category. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)* (pp. 465–470).

Micucci, D., Mobilio, M., & Napoletano, P. (2016). UniMiB SHAR: a new dataset for human activity recognition using acceleration data from smartphones. *ArXiv e-prints*. Retrieved from https://arxiv.org/abs/1611.07688

*Minnen, D., Starner, T., Essa, I., & Isbell, C. (2006). Discovering Characteristic Actions from On-Body Sensor Data. *Wearable Computers, 2006 10th IEEE International Symposium on*, 11–18.

*Minnen, D., Westeyn, T., Ashbrook, D., Presti, P., & Starner, T. (2007). Recognizing soldier activities in the field. In *4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007)* (pp. 236–241). Springer.

*Mitchell, E., Monaghan, D., & O'Connor, N. (2013). Classification of Sporting Activities Using Smartphone Accelerometers. *Sensors*, *13*(4), 5317–5337. Retrieved 2014-06-03, from http://www.mdpi.com/1424-8220/13/4/5317/ doi: 10.3390/s130405317

Mizell, D., & Cray, I. (2003). Using gravity to estimate accelerometer orientation. In *Wearable Computers, 2003 7th IEEE International Symposium on. (ISWC* (Vol. 1530, p. 17). Citeseer.

*Moder, T., Hafner, P., Wisiol, K., & Wieser, M. (2014). 3d indoor positioning with pedestrian dead reckoning and activity recognition based on bayes filtering. In *Indoor positioning and indoor navigation (IPIN), 2014 international conference on* (pp. 717–720). IEEE.

Moeslund, T. B., & Granum, E. (2001). A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, *81*(3), 231–268. Retrieved 2012-08-24, from http://www.sciencedirect.com/science/article/pii/S107731420090897X doi: 10.1006/cviu.2000.0897

Mollineda, R. A., Alejo, R., & Sotoca, J. M. (2007). The class imbalance problem in pattern classification and learning. In *II Congreso Español de Informática (CEDI 2007). ISBN* (pp. 978–84).

Moore, S. T., MacDougall, H. G., & Ondo, W. G. (2008). Ambulatory monitoring of freezing of gait in parkinson's disease. *Journal of Neuroscience Methods*, *167*(2), 340–348. doi: 10.1016/j.jneumeth.2007.08.023

*Moufawad el Achkar, C., Lenoble-Hoskovec, C., Paraschiv-Ionescu, A., Major, K., Büla, C., & Aminian, K. (2016). Instrumented shoes for activity classification in the elderly. *Gait & Posture*, *44*, 12–17. doi: 10.1016/j.gaitpost.2015.10.016

*Najafi, B., Aminian, K., Paraschiv-Ionescu, A., Loew, F., Bula, C., & Robert, P. (2003). Ambulatory system for human motion analysis using a kinematic sensor: monitoring of daily physical activity in the elderly. *IEEE Transactions on Biomedical Engineering*, *50*(6), 711–723. doi: 10.1109/TBME.2003.812189

*Nguyen, A., Moore, D., & McCowan, I. (2007). Unsupervised clustering of free-living human activities using ambulatory accelerometry. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE* (pp. 4895–4898). IEEE.

Ni, B., Wang, G., & Moulin, P. (2013). Rgbd-hudaact: A color-depth video database for human daily activity recognition. In *Consumer depth cameras for computer vision* (pp. 193–208). Springer. Retrieved 2015-01-27, from http://www.ntu.edu.sg/home/wanggang/NiWangMoulin2011.pdf

Ni, T. (2011). *A Framework of Freehand Gesture Interaction: Techniques, Guidelines, and Applications* (PhD, Virginia Polytechnic Institute and State University, Blacksburg, Virginia). Retrieved 2012-10-10, from http://scholar.lib.vt

.edu/theses/available/etd-09212011-230923/

Niebles, J. C., Chen, C.-W., & Fei-Fei, L. (2010). Modeling Temporal Structure of Decomposable Motion Segments for Activity Classification. In K. Daniilidis, P. Maragos, & N. Paragios (Eds.), *Computer Vision ECCV 2010* (pp. 392–405). Springer Berlin Heidelberg.

*Noor, M. H. M., Salcic, Z., & Wang, K. I. (2016). Adaptive sliding window segmentation for physical activity recognition using a single tri-axial accelerometer. *Pervasive and Mobile Computing*. doi: 10.1016/j.pmcj.2016.09.009

Nordlie, E., Gewaltig, M.-O., & Plesser, H. E. (2009). Towards reproducible descriptions of neuronal network models. *PLoS Comput Biol*, *5*(8), e1000456. doi: 10.1371/journal.pcbi.1000456

Novatchkov, H., & Baca, A. (2012). Machine learning methods for the automatic evaluation of exercises on sensor-equipped weight training machines. *Procedia Engineering*, *34*, 562–567. Retrieved 2014-06-03, from http://www.sciencedirect.com/science/article/pii/S1877705812017092 doi: 10.1016/j.proeng.2012.04.096

*Nyan, M., Tay, F. E., & Murugasu, E. (2008). A wearable system for pre-impact fall detection. *Journal of Biomechanics*, *41*(16), 3475–3481. doi: 10.1016/j.jbiomech.2008.08.009

Ogris, G., Stiefmeier, T., Junker, H., Lukowicz, P., & Troster, G. (2005). Using ultrasonic hand tracking to augment motion analysis based recognition of manipulative gestures. In *Wearable Computers, 2005. Proceedings. Ninth IEEE International Symposium on* (pp. 152–159). IEEE. Retrieved 2015-08-06, from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1550800

*Oniga, S., & Suto, J. (2014). Human activity recognition using neural networks. In *Control conference (iccc), 2014 15th international carpathian* (pp. 403–406). IEEE. doi: 10.1109/CarpathianCC.2014.6843636

Opportunity Consortium. (2016). *Opportunity challenge - tasks opportunity.* Retrieved from http://www.opportunity-project.eu/node/48

*Palumbo, F., Gallicchio, C., Pucci, R., & Micheli, A. (2016). Human activity recognition using multisensor data fusion based on reservoir computing. *Journal of Ambient Intelligence and Smart Environments*, *8*(2), 87–107. doi: 10.3233/AIS-160372

Pantic, M., Pentland, A., Nijholt, A., & Huang, T. (2007). Human computing and machine understanding of human behavior: A survey. In T. Huang, A. Nijholt, M. Pantic, & A. Pentland (Eds.), *Artifical intelligence for human computing* (Vol. 4451, pp. 47–71). Springer Berlin / Heidelberg.

*Paraschiv-Ionescu, A., Buchser, E., Rutschmann, B., Najafi, B., & Aminian, K. (2004). Ambulatory system for the quantitative and qualitative analysis of gait and posture in chronic pain patients treated with spinal cord stimulation. *Gait & Posture*, *20*(2), 113–125. doi: 10.1016/j.gaitpost.2003.07.005

*Parkka, J., Ermes, M., Korpipaa, P., Mantyjarvi, J., Peltola, J., & Korhonen, I. (2006). Activity Classification Using Realistic Data From Wearable Sensors. *IEEE Transactions on Information Technology in Biomedicine*, *10*(1), 119–128. doi: 10.1109/TITB.2005.856863

Patterson, S., Krantz, D. S., Montgomery, L. C., Deuster, P. A., Hedges, S. M., & Nebel, L. E. (1993). Automated physical activity monitoring: Validation and comparison with physiological and self-report measures. *Psychophysiology*, *30*(3), 296–306.

*Pavey, T. G., Gilson, N. D., Gomersall, S. R., Clark, B., & Trost, S. G. (2017). Field evaluation of a random forest activity classifier for wrist-worn accelerometer data. *Journal of Science and Medicine in Sport*, *20*(1), 75–80. doi: 10.1016/j.jsams.2016.06.003

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007, December). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *24*(3), 45–77. Retrieved 2015-12-18, from http://www.tandfonline.com/doi/full/10.2753/MIS0742-1222240302 doi: 10.2753/MIS0742-1222240302

Perng, J. K., Fisher, B., Hollar, S., & Pister, K. S. (1999). Acceleration sensing glove. In *16th international symposium on wearable computers* (pp. 178–178). IEEE Computer Society. Retrieved 2015-01-30, from http://www.computer.org/csdl/proceedings/iswc/1999/0428/00/04280178.pdf

*Peterek, T., Penhaker, M., Gajdous, P., & Dohnalek, P. (2014). Comparison of classification algorithms for physical activity recognition. In *Innovations in bio-inspired computing and applications* (pp. 123–131). Springer International Publishing. (DOI: 10.1007/978-3-319-01781-5_12)

*Pham, C., Plotz, T., & Olivier, P. (2010). A dynamic time warping approach to real-time activity recognition for food preparation. In *Ambient Intelligence* (pp. 21–30). Springer.

*Pham, N., & Abdelzaher, T. (2008). Robust dynamic human activity recognition based on relative energy allocation. In *Distributed Computing in Sensor Systems* (pp. 525–530). Springer.

*Pirttikangas, S., Fujinami, K., & Nakajima, T. (2006). Feature selection and activity recognition from wearable sensors. In *Ubiquitous Computing Systems* (pp. 516–527). Springer.

Plötz, T. (2010). How to do good research in activity recognition. In *How to do good research in activity recognition; workshop in conjunction with pervasive 2010* (p. 4). Heksinki, Finland.

*Pober, D. M., Staudenmayer, J., Raphael, C., & Freedson, P. S. (2006). Development of Novel Techniques to Classify Physical Activity Mode Using Accelerometers:. *Medicine & Science in Sports & Exercise*, *38*(9), 1626–1634. doi: 10.1249/01.mss.0000227542.43669.45

Prater, A. (2016). Reservoir computing for spatiotemporal signal classification without trained output weights. *arXiv preprint arXiv:1604.03073*.

Preece, S., Goulermas, J., Kenney, L. P. J., & Howard, D. (2009). A Comparison of Feature Extraction Methods for the Classification of Dynamic Activities From Accelerometer Data. *IEEE Transactions on Biomedical Engineering*, *56*(3), 871–879. doi: 10.1109/TBME.2008.2006190

*Preece, S. J., Goulermas, J. Y., Kenney, L. P. J., Howard, D., Meijer, K., & Crompton, R. (2009). Activity identification using body-mounted sensors – a re-

view of classification techniques. *Physiological Measurement*, *30*(4), 33. doi: 10.1088/0967-3334/30/4/R01

Psarrou, A., Gong, S., & Walter, M. (2002). Recognition of human gestures and behaviour based on motion trajectories. *Image and Vision Computing*, *20*(5-6), 349–358.

Qaisar, S., Imtiaz, S., Faruq, F., Jamal, A., Iqbal, W., Glazier, P., & Lee, S. (2013). A hidden markov model for detection and classification of arm action in cricket using wearable sensors. *Journal of Mobile Multimedia*, *9*(1&2), 128–144. Retrieved 2014-06-03, from http://www.paulglazier.info/images/pdfs/qaisar_et_al_13b.pdf

R Core Team. (2012). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from http://www.R-project.org/

Ramamoorthy, A., Vaswani, N., Chaudhury, S., & Banerjee, S. (2003). Recognition of dynamic hand gestures. *Pattern Recognition*, *36*(9), 2069–2081.

*Randell, C., & Muller, H. (2000). Context awareness by analysing accelerometer data. In *Wearable computers, 2000 4th IEEE international symposium on* (Vol. 1, pp. 175–176). IEEE Computer Society. doi: 10.1109/ISWC.2000.888488

*Ravi, N., Dandekar, N., Mysore, P., & Littman, M. (2005). Activity recognition from accelerometer data. In *Artificial Intelligence, Seventeenth Conference on Innovative Applications of. (IAAI).*

*Reiss, A., Hendeby, G., & Stricker, D. (2013). A competitive approach for human activity recognition on smartphones. In *Proceedings / 21st european symposium on artificial neural networks, computational intelligence and machine learning, ESANN 2013* (pp. 455–460).

Reiss, A., & Stricker, D. (2012). Introducing a new benchmarked dataset for activity monitoring. In *Proceedings of the 16th international symposium on wearable computers* (pp. 108–109). IEEE. doi: 10.1109/ISWC.2012.13

*Ren, X., Ding, W., Crouter, S. E., Mu, Y., & Xie, R. (2016). Activity recognition and intensity estimation in youth from accelerometer data aided by machine learning. *Applied Intelligence*, *45*(2), 512–529. doi: 10.1007/s10489-016-0773-3

Revelle, W. (2014). psych: Procedures for psychological, psychometric, and personality research [Computer software manual]. Evanston, Illinois. Retrieved from http://CRAN.R-project.org/package=psych

*Robert, B., White, B., Renter, D., & Larson, R. (2009). Evaluation of three-dimensional accelerometers to monitor and classify behavior patterns in cattle. *Computers and Electronics in Agriculture*, *67*(1–2), 80–84. doi: 10.1016/j.compag.2009.03.002

Roetenberg, D., Slycke, P., & Veltink, P. (2007). Ambulatory position and orientation tracking fusing magnetic and inertial sensing. *Biomedical Engineering, IEEE Transactions on*, *54*(5), 883–890.

Roh, M.-C., Christmas, B., Kittler, J., & Lee, S.-W. (2008). Gesture spotting for low-resolution sports video annotation. *Pattern Recognition*, *41*(3), 1124–1137.

*Ronao, C. A., & Cho, S. (2014). Human activity recognition using smartphone sensors with two-stage continuous hidden markov models. In *Natural computation (ICNC), 2014 10th international conference on* (pp. 681–686). IEEE.

*Salarian, A., Russmann, H., Vingerhoets, F., Burkhard, P., & Aminian, K. (2007). Ambulatory Monitoring of Physical Activities in Patients With Parkinson's Disease. *IEEE Transactions on Biomedical Engineering*, *54*(12), 2296–2299. doi: 10.1109/TBME.2007.896591

Schiller, U. D., & Steil, J. J. (2005). Analyzing the weight dynamics of recurrent learning algorithms. *Neurocomputing*, *63*, 5–23.

Schliebs, S., Defoin-Platel, M., & Kasabov, N. (2009). Integrated feature and parameter optimization for an evolving spiking neural network. In M. Köppen, N. K. Kasabov, & G. G. Coghill (Eds.), *Advances in Neuro-Information Processing, 15th International Conference* (Vol. 5506, pp. 1229–1236). Heidelberg, Germany: Springer. doi: 10.1007/978-3-642-02490-0_149

Schliebs, S., Fiasché, M., & Kasabov, N. (2012). Constructing robust liquid state machines to process highly variable data streams. In *Icann'12* (pp. 604–612). Lausanne, Switzerland: Springer.

Schliebs, S., & Hunt, D. P. L. (2012). Continuous classification of spatio-temporal data streams using liquid state machines. In *Iconip'12* (pp. 626–633). Qatar: Springer.

Schliebs, S., Kasabov, N., Parry, D., & Hunt, D. P. L. (2013). Towards a wearable coach: Classifying sports activities with reservoir computing. In C. J. L Iliadis H Papadopoulos (Ed.), *Engineering applications of neural networks* (Vol. 383, pp. 233–242). Springer.

Schlomer, T., Poppinga, B., Henze, N., & Boll, S. (2008). Gesture recognition with a Wii controller. In *Proceedings of the 2nd international conference on Tangible and embedded interaction* (pp. 11–14). Bonn, Germany: ACM.

Schrauwen, B., D'Haene, M., Verstraeten, D., & Campenhout, J. V. (2008). Compact hardware liquid state machines on fpga for real-time speech recognition. *Neural Networks*, *21*(2–3), 511–523. doi: 10.1016/j.neunet.2007.12.009

Schrauwen, B., & Van Campenhout, J. (2003). BSA, a fast and accurate spike train encoding scheme. In *Neural networks, 2003. proceedings of the international joint conference on* (Vol. 4, pp. 2825–2830). doi: 10.1109/IJCNN.2003.1224019

*Sekine, M., Tamura, T., Fujimoto, T., & Fukui, Y. (2000). Classification of walking pattern using acceleration waveform in elderly people. In *Engineering in Medicine and Biology Society, 2000. Proceedings of the 22nd Annual International Conference of the IEEE* (Vol. 2, pp. 1356–1359). IEEE.

Sheik, S., Coath, M., Indiveri, G., Denham, S. L., Wennekers, T., & Chicca, E. (2012). Emergent Auditory Feature Tuning in a Real-Time Neuromorphic VLSI System. *Frontiers in Neuroscience*, *6*. doi: 10.3389/fnins.2012.00017

*Shoaib, M., Scholten, H., & Havinga, P. J. M. (2013). Towards physical activity recognition using smartphone sensors. In *Ubiquitous intelligence and computing, 2013 ieee 10th international conference on and 10th international conference on autonomic and trusted computing (uic/atc)* (pp. 80–87). IEEE. doi:

10.1109/UIC-ATC.2013.43

Smith, C. S. (1999). Activities: States or Events? *Linguistics and Philosophy*, *22*(5), 479–508. doi: 10.1023/A:1005439826271

Soh, H., & Demiris, Y. (2014). Incrementally learning objects by touch: Online discriminative and generative models for tactile-based recognition. *IEEE Transactions on Haptics*, *Early Access Online*, 10. doi: 10.1109/TOH.2014.2326159

SparkFun Electronics Inc. (2008a). *IMU 6 degrees of freedom - v4 with bluetooth capability.* Product Description. Retrieved from https://www.sparkfun.com/products/retired/8454

SparkFun Electronics Inc. (2008b). *SparkFun 6DoF v4 imu noise problems.* Web Support Note. Retrieved from https://www.sparkfun.com/products/retired/8454

Spelmezan, D., Schanowski, A., & Borchers, J. (2009). Wearable automatic feedback devices for physical activities. In *Proceedings of the fourth international conference on body area networks* (pp. 1:1–1:8). Brussels, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). doi: 10.4108/ICST.BODYNETS2009.6095

Spriggs, E. H., De La Torre, F., & Hebert, M. (2009). Temporal segmentation and activity classification from first-person sensing. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference On* (pp. 17–24). IEEE.

*Steele, B. G., Belza, B., Cain, K., Warms, C., Coppersmith, J., & Howard, J. (2003). Bodies in motion: monitoring daily activity and exercise with motion sensors in people with chronic pulmonary disease. *Journal of rehabilitation research and development*, *40*(5; SUPP/2), 45–58.

Steil, J. J. (2004). Backpropagation-decorrelation: online recurrent learning with o (n) complexity. In *Neural networks, 2004. proceedings. 2004 IEEE international joint conference on* (Vol. 2, pp. 843–848). Retrieved 2015-01-16, from http://cognitionandrobotics.de/system/files/u9/Steil_BPDC_IJCNN2004.pdf

*Stiefmeier, T., Ogris, G., Junker, H., Lukowicz, P., & Troster, G. (2006). Combining motion sensors and ultrasonic hands tracking for continuous activity recognition in a maintenance scenario. In *Wearable computers, 2006 10th ieee international symposium on* (pp. 97–104).

Stiefmeier, T., Roggen, D., Ogris, G., Lukowicz, P., & Troster, G. (2008). Wearable activity tracking in car manufacturing. *IEEE Pervasive Computing*, *7*(2), 42–50.

*Stikic, M., Larlus, D., Ebert, S., & Schiele, B. (2011). Weakly Supervised Recognition of Daily Life Activities with Wearable Sensors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(12), 2521–2537. doi: 10.1109/TPAMI.2011.36

*Stikic, M., Larlus, D., & Schiele, B. (2009). Multi-graph based semi-supervised learning for activity recognition. In *Wearable Computers, 2009. ISWC'09. International Symposium on* (pp. 85–92). IEEE.

*Stikic, M., Van Laerhoven, K., & Schiele, B. (2008). Exploring semi-supervised and active learning for activity recognition. In *Wearable Computers, 2008. ISWC*

2008. 12th IEEE International Symposium on (pp. 81–88). IEEE.

*Stisen, A., Blunck, H., Bhattacharya, S., Prentow, T. S., Kjærgaard, M. B., Dey, A., ... Jensen, M. M. (2015). Smart devices are different: Assessing and MitigatingMobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th acm conference on embedded networked sensor systems* (pp. 127–140). ACM Press. doi: 10.1145/2809695.2809718

Stodden, V., Leisch, F., & Peng, R. D. (2014). *Implementing reproducible research*. Chapman and Hall/CRC.

*Sztyler, T., Stuckenschmidt, H., & Petrich, W. (2017). Position-aware activity recognition with wearable devices. *Pervasive and Mobile Computing*. doi: 10.1016/j.pmcj.2017.01.008

Tang, Y., Zhang, Y., Chawla, N. V., & Krasser, S. (2009). SVMs Modeling for Highly Imbalanced Classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *39*(1), 281–288. doi: 10.1109/TSMCB.2008.2002909

Tangient LLC. (2016). *BoxLab - list of home datasets.* Retrieved from http://boxlab.wikispaces.com/List+of+Home+Datasets

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

Taylor, K., Abdulla, U. A., Helmer, R. J. N., Lee, J., & Blanchonette, I. (2011). Activity classification with smart phones for sports activities. *Procedia Engineering*, *13*, 428–433. Retrieved 2014-06-03, from http://www.sciencedirect.com/science/article/pii/S187770581101023X doi: 10.1016/j.proeng.2011.05.109

*Uiterwaal, M., Glerum, E. B. C., Busser, H. J., & Van Lummel, R. C. (1998). Ambulatory monitoring of physical activity in working situations, a validation study. *Journal of medical engineering & technology*, *22*(4), 168–172.

Vahdatpour, A., Amini, N., & Sarrafzadeh, M. (2009). Toward unsupervised activity discovery using multi-dimensional motif detection in time series. In *Artifical intelligence, proceedings of the 21st international joint conference on* (pp. 1261–1266).

Vaishnavi, V. K., & Kuechler, W. (2015). *Design science research methods and patterns: innovating information and communication technology*. Crc Press.

van Kasteren, T., Englebienne, G., & Krose, B. (2011). Human activity recognition from wireless sensor network data: Benchmark and software. In *Activity recognition in pervasive intelligent environments* (pp. 165–186). Springer.

*Van Laerhoven, K., & Cakmakci, O. (2000). What shall we teach our pants. In *Wearable computers, 2000 4th IEEE international symposium on* (pp. 77–83). Citeseer. doi: 10.1109/ISWC.2000.888468

*van Laerhoven, K., & Gellersen, H.-W. (2004). Spine versus porcupine: A study in distributed wearable activity recognition. In *Wearable Computers, 2004. ISWC 2004. Eighth International Symposium on* (Vol. 1, pp. 142–149). IEEE.

*van Laerhoven, K., Gellersen, H.-W., & Malliaris, Y. G. (2006). Long term activity monitoring with a wearable sensor node. In *Wearable and Implantable Body*

Sensor Networks, 2006. BSN 2006. International Workshop on (p. 4). IEEE.

Vavoulas, G., Pediaditis, M., Spanakis, E. G., & Tsiknakis, M. (2013). The mobifall dataset: An initial evaluation of fall detection algorithms using smartphones. In *Bioinformatics and bioengineering (BIBE), 2013 IEEE 13th international conference on* (pp. 1–4). IEEE.

*Veltink, P. H., Bussmann, H. J., de Vries, W., Martens, W. J., & Van Lummel, R. C. (1996). Detection of static and dynamic activities using uniaxial accelerometers. *Rehabilitation Engineering, IEEE Transactions on*, *4*(4), 375–385.

Venayagamoorthy, G. K. (2007). Online design of an echo state network based wide area monitor for a multimachine power system. *Neural Networks*, *20*(3), 404–413. Retrieved 2014-06-09, from http://www.sciencedirect.com/science/article/pii/S0893608007000500 doi: 10.1016/j.neunet.2007.04.021

Verplaetse, C. (1996). Inertial proprioceptive devices: self-motion-sensing toys and tools. *IBM Systems Journal*, *35*(3), 639–650.

Verstraeten, D., Schrauwen, B., D'Haene, M., & Stroobandt, D. (2007). An experimental unification of reservoir computing methods. *Neural Networks*, *20*(3), 391–403. doi: 10.1016/j.neunet.2007.04.003

Verstraeten, D., Schrauwen, B., & Stroobandt, D. (2005). Isolated word recognition using a liquid state machine. In *ESANN* (pp. 435–440).

Verstraeten, D., Schrauwen, B., & Stroobandt, D. (2006). Reservoir-based techniques for speech recognition. In *Proceedings of the international joint conference on neural networks, ijcnn* (p. 1050-1053).

*Vinh, L. T., Lee, S., Le, H. X., Ngo, H. Q., Kim, H. I., Han, M., & Lee, Y. (2011). Semi-Markov conditional random fields for accelerometer-based activity recognition. *Applied Intelligence*, *35*(2), 226–241. doi: 10.1007/s10489-010-0216-5

*Vital, J. P. M., Faria, D. R., Dias, G., Couceiro, M. S., Coutinho, F., & Ferreira, N. M. F. (2016). Combining discriminative spatiotemporal features for daily life activity recognition using wearable motion sensing suit. *Pattern Analysis and Applications*. doi: 10.1007/s10044-016-0558-7

*Vollmer, C., Gross, H., & Eggert, J. P. (2013). Learning features for activity recognition with shift-invariant sparse coding. In *International conference on artificial neural networks* (pp. 367–374). Springer.

*Walker, D. J., Heslop, P. S., Plummer, C. J., Essex, T., & Chandler, S. (1997). A continuous patient activity monitor: validation and relation to disability. *Physiological Measurement*, *18*(1), 49. doi: 10.1088/0967-3334/18/1/003

Walter, P. L. (2007). The history of the accelerometer: 1920s-1996 – prologue and epilogue, 2006. *Sound and Vibration*, *41*(1), 84.

*Wang, N., Ambikairajah, E., Lovell, N. H., & Celler, B. G. (2007). Accelerometry based classification of walking patterns using time-frequency analysis. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE* (pp. 4899–4902). IEEE.

*Ward, J., Lukowicz, P., Troster, G., & Starner, T. E. (2006). Activity recognition of assembly tasks using body-worn microphones and accelerometers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *28*(10), 1553–1567.

Ward, J. A., Lukowicz, P., & Troster, G. (2006). Evaluating performance in continuous context recognition using event-driven error characterisation. In *Location and Context-Awareness* (pp. 239–255). Springer.

Weiser, M. (1993). Some computer science issues in ubiquitous computing. *Communications of the ACM*, *36*(7), 75–84. doi: 10.1145/159544.159617

Weiss, G. M. (2004). Mining with rarity: a unifying framework. *ACM SIGKDD Explorations Newsletter*, *6*(1), 7–19.

Weiss, G. M. (2010). Mining with rare cases. In O. Maimon & L. Rokach (Eds.), *Data mining and knowledge discovery handbook* (pp. 747–757). Springer US. Retrieved 2012-08-21, from http://www.springerlink.com.ezproxy.aut.ac.nz/content/m691323x28002vk7/abstract/

*Weiss, G. M., & Lockhart, J. W. (2012). The impact of personalization on smartphone-based activity recognition. In *Activity context representation: Techniques and languages AAAI technical report WS-12-05.*

Welch, G., & Foxlin, E. (2002). Motion Tracking: No Silver Bullet, but a Respectable Arsenal. *Computer Graphics and Applications, IEEE*, *22*(6), 24–38. doi: 10.1109/MCG.2002.1046626

*Wenlong, T., & Sazonov, E. S. (2014). Highly accurate recognition of human postures and activities through classification with rejection. *IEEE Journal of Biomedical and Health Informatics*, *18*(1), 309–315. doi: 10.1109/JBHI.2013.2287400

Westeyn, T., Brashear, H., Atrash, A., & Starner, T. (2003). Georgia tech gesture toolkit: Supporting experiments in gesture recognition. In *Multimodal interfaces, International conference on - 5th* (pp. 85–92). ACM New York, NY, USA.

Winter, R. (2008). Design science research in Europe. *European Journal of Information Systems*, *17*(5), 470–475. doi: 10.1057/ejis.2008.44

Wixted, A., Portus, M., Spratford, W., & James, D. (2011). Detection of throwing in cricket using wearable sensors. *Sports Technology*, *4*(3-4), 134–140. Retrieved 2014-06-03, from http://www.tandfonline.com/doi/abs/10.1080/19346182.2012.725409 doi: 10.1080/19346182.2012.725409

*Wu, G., & Xue, S. (2008). Portable Preimpact Fall Detector With Inertial Sensors. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *16*(2), 178–183. doi: 10.1109/TNSRE.2007.916282

*Wu, W. H., Bui, A. A. T., Batalin, M. A., Au, L. K., Binney, J. D., & Kaiser, W. J. (2008). MEDIC: Medical embedded device for individualized care. *Artificial Intelligence in Medicine*, *42*(2), 137–152. doi: 10.1016/j.artmed.2007.11.006

Wyatt, D., Philipose, M., & Choudhury, T. (2005). Unsupervised activity recognition using automatically mined common sense. In *AAAI 05* (Vol. 5, pp. 21–27).

Wynekoop, J. L., & Conger, S. (1990). A review of computer aided software engineering research methods. *Information Systems Research Arena of the 90's, The: Challenges, Perceptions, and Alternative Approaches*, *1*, 129–154.

Yang, A. Y., Jafari, R., Sastry, S. S., & Bajcsy, R. (2009). Distributed recognition of human actions using wearable motion sensor networks. *Journal of Ambient Intelligence and Smart Environments*, *1*(2), 103–115.

Yang, C., & Hsu, Y. (2010). A review of accelerometry-based wearable motion detectors for physical activity monitoring. *Sensors*, *10*(8), 7772–7788. doi: 10.3390/s100807772

*Yang, J., & Wang, Y., J.and Chen. (2008). Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers. *Pattern Recognition Letters*, *29*(16), 2213–2220. doi: 10.1016/j.patrec.2008.08.002

*Yin, J., Yang, Q., & Pan, J. J. (2008). Sensor-Based Abnormal Human-Activity Detection. *IEEE Transactions on Knowledge and Data Engineering*, *20*(8), 1082–1090. doi: 10.1109/TKDE.2007.1042

Yoon, H., Soh, J., Bae, Y. J., & Yang, H. S. (2001). Hand gesture recognition using combined features of location, angle and velocity. *Pattern Recognition*, *34*(7), 1491–1501.

YOST Labs. (2015). *YEI Technology – 3-Space AHRS IMU Sensors.* Retrieved 2016-01-15, from https://www.yeitechnology.com/product-category/3-space-sensor

*Zappi, P., Lombriser, C., Stiefmeier, T., Farella, E., Roggen, D., Benini, L., & Troster, G. (2008). Activity recognition from on-body sensors: accuracy-power trade-off by dynamic sensor selection. In *Wireless sensor networks* (pp. 17–33). Springer. Retrieved 2015-01-23, from http://link.springer.com/chapter/10.1007/978-3-540-77690-1_2

Zappi, P., Stiefmeier, T., Farella, E., Roggen, D., Benini, L., & Troster, G. (2007). Activity recognition from on-body sensors by classifier fusion: sensor scalability and robustness. *ISSNIP, Proceedings of*, *1*, 281–286. doi: 10.1109/ISSNIP.2007.4496857

Zawodny, J. (2007). *Boat wake patterns in water.* Retrieved 2012-08-09, from http://www.flickr.com/photos/jzawodn/1325529844/ (Copyright 2007 by Jeremy Zawodny. Reprinted under license, see: http://creativecommons.org/licenses/by-nc/2.0/deed.en)

*Zhang, H. (2006). *Control Freaks* (Unpublished doctoral dissertation). Interaction Design Institute Ivrea, Milan.

Zhang, H., Zhou, W., & Parker, L. E. (2014). Fuzzy segmentation and recognition of continuous human activities. In *IEEE international conference on robotics and automation (ICRA)* (pp. 6305–6312). IEEE.

Zhang, M., & Sawchuk, A. A. (2012). USC-HAD: a daily activity dataset for ubiquitous activity recognition using wearable sensors. In *Proceedings of the 2012 ACM conference on ubiquitous computing* (pp. 1036–1043). ACM.

*Zhang, S., Mccullagh, P., & Callaghan, V. (2014). An Efficient Feature Selection Method for Activity Classification. In *Proceedings of the international conference on intelligent environments* (pp. 16–22). IEEE. doi: 10.1109/IE.2014.10

*Zhang, T., Wang, J., Xu, L., & Liu, P. (2006). Fall detection by wearable sensor and one-class SVM algorithm. In *Intelligent Computing in Signal Processing and Pattern Recognition* (pp. 858–863). Springer.

*Zhao, Z., Chen, Z., Chen, Y., Wang, S., & Wang, H. (2014). A class incremental extreme learning machine for activity recognition. *Cognitive Computation*,

6(3), 423–431. doi: 10.1007/s12559-014-9259-y

Zhou, H., & Hu, H. (2005). Kinematic model aided inertial motion tracking of human upper limb. In *Information Acquisition, IEEE International Conference on - 2005.* Hong Kong and Macau, China.

Zhou, H., & Hu, H. (2008). Human motion tracking for rehabilitation – A survey. *Biomedical Signal Processing and Control*, *3*(1), 1–18. doi: 10.1016/j.bspc .2007.09.001

Zhou, H., Stone, T., Hu, H., & Harris, N. (2008). Use of multiple wearable inertial sensors in upper limb motion tracking. *Medical Engineering & Physics*, *30*(1), 123–133. doi: 10.1016/j.medengphy.2006.11.010

Zhou, Z., & Liu, X. (2006). Training cost-sensitive neural networks with methods addressing the class imbalance problem. *Knowledge and Data Engineering, IEEE Transactions on*, *18*(1), 63–77.

Zhu, R., & Zhou, Z. (2004). A real-time articulated human motion tracking using tri-axis inertial/magnetic sensors package. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, *12*(2), 295–302.

*Zinnen, A., Blanke, U., & Schiele, B. (2009). An Analysis of Sensor-Oriented vs. Model-Based Activity Recognition. In *Wearable Computers, 2009 International Symposium on. ISWC '09.* (pp. 93–100). doi: 10.1109/ISWC.2009.32

Zinnen, A., & Schiele, B. (2008). A new Approach to Enable Gesture Recognition in Continuous Data Streams. In *Wearable Computers, 2008 International Symposium on. ISWC '08.* (pp. 33–40). doi: 10.1109/ISWC.2008.4911581

ACRONYMS

ACS Activity Classification System. 1, 6, 11, 14, 16, 17, 20, 22, 27, 47, 67, 71, 236
AmI Ambient Intelligence. 17
AUC Area Under the Curve. 168

BD Backpropagation Decorrelation. 52
BSA Ben's Spike Algorithm. 98, 99, 101, 239

DTW Dynamic Time Warp. 63
DWT Discrete Wavelet Transform. 32

ESN Echo State Network. 6, 7, 9, 51, 52, 53, 54, 55, 56, 58, 73, 74, 86, 88, 90, 91, 94, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 128, 130, 132, 133, 134, 135, 136, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 158, 159, 160, 161, 162, 163, 164, 168, 169, 170, 171, 172, 173, 174, 175, 178, 179, 180, 181, 182, 187, 188, 190, 191, 192, 193, 197, 198, 199, 200, 202, 205, 206, 207, 209, 211, 212, 214, 217, 219, 221, 222, 223, 224, 225, 226, 231, 232, 233, 234, 236, 237, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 251, 252, 258, 260, 263, 265, 266, 267, 268, 269, 271, 273, 274

FFT Fast Fourier Transform. 32

HCI Human Computer Interface. 47
HMM Hidden Markov Model. 16, 25, 31, 36, 43, 46, 55, 57, 240, 274

ICA Independent Component Analysis. 23
IMU Inertial Measurement Unit. 9, 23, 76, 77, 83, 206, 255, 256, 257, 274

KEDRI Knowledge Engineering and Discovery Research Institute. 121

LIF Leaky Integrate-and-Fire. 52, 96, 102, 115
LSM Liquid State Machine. 6, 7, 9, 51, 52, 53, 55, 56, 73, 94, 95, 96, 99, 101, 102, 104, 108, 110, 111, 112, 113, 114, 115, 119, 120, 121, 122, 123, 124, 126, 128, 140, 198, 239, 240, 271, 274

PCA Principal Component Analysis. 23
PSO Particle Swarm Optimiser. 7, 114, 117, 118, 121, 123, 124, 125, 132, 133, 134, 140, 144, 151, 152, 153, 161, 171, 172, 179, 180, 231, 238, 240

RC Reservoir Computing. 1, 4, 5, 6, 7, 27, 51, 52, 55, 56, 57, 58, 67, 68, 69, 70, 71, 72, 73, 94, 95, 111, 112, 113, 120, 121, 122, 123, 143, 236, 237, 239, 240, 242, 253, 258, 259, 260, 261, 265, 266, 271, 272, 273, 274, 275
RMSE Root Mean Square Error. 214, 217, 219, 226

RNN Recurrent Neural Network. 9, 10, 46, 51, 52, 55, 71, 262
ROC Receiver Operating Characteristic. 4, 5, 142, 158, 159, 160, 164, 165, 168, 169, 173, 175, 178, 191, 193, 196, 198, 199, 202, 205, 206, 207, 209, 214, 217, 219, 221, 224, 226, 227, 231, 241, 242, 243, 244, 245, 246, 247, 248, 249, 251, 252
RSNN Recurrent Spiking Neural Network. 52, 53, 95, 97

SNN Spiking Neural Network. 95, 96, 98, 102, 107
STP Short-Term Plasticity. 52, 96, 102
SVM Support Vector Machine. 56, 60, 125, 240, 259, 262, 273, 274
SWI Small-world inter-connectivity. 96, 102

VR Virtual Reality. 23