



PHD

**Machine Learning Methods for Locomotion Mode Recognition with Application to Active Lower Limb Prosthesis
(Alternative Format Thesis)**

Sherratt, Freddie

Award date:
2022

Awarding institution:
University of Bath

[Link to publication](#)

Alternative formats

If you require this document in an alternative format, please contact:
openaccess@bath.ac.uk

Copyright of this thesis rests with the author. Access is subject to the above licence, if given. If no licence is specified above, original content in this thesis is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC-ND 4.0) Licence (<https://creativecommons.org/licenses/by-nc-nd/4.0/>). Any third-party copyright material present remains the property of its respective owner(s) and is licensed under its existing terms.

Take down policy

If you consider content within Bath's Research Portal to be in breach of UK law, please contact: openaccess@bath.ac.uk with the details. Your claim will be investigated and, where appropriate, the item will be removed from public view as soon as possible.



PHD

**Machine Learning Methods for Locomotion Mode Recognition with Application to Active Lower Limb Prosthesis
(Alternative Format Thesis)**

Sherratt, Freddie

Award date:
2022

Awarding institution:
University of Bath

[Link to publication](#)

Alternative formats

If you require this document in an alternative format, please contact:
openaccess@bath.ac.uk

Copyright of this thesis rests with the author. Access is subject to the above licence, if given. If no licence is specified above, original content in this thesis is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC-ND 4.0) Licence (<https://creativecommons.org/licenses/by-nc-nd/4.0/>). Any third-party copyright material present remains the property of its respective owner(s) and is licensed under its existing terms.

Take down policy

If you consider content within Bath's Research Portal to be in breach of UK law, please contact: openaccess@bath.ac.uk with the details. Your claim will be investigated and, where appropriate, the item will be removed from public view as soon as possible.

**Machine Learning Methods for
Locomotion Mode Recognition with
Application to Active Lower Limb
Prosthesis**

submitted by

Frederick W Sherratt

for the degree of Doctor of Philosophy

of the

University of Bath

Department of Mechanical Engineering

February 2022

Copyright Notice

Attention is drawn to the fact that copyright of this thesis/portfolio rests with the author and copyright of any previously published materials included may rest with third parties. A copy of this thesis/portfolio has been supplied on condition that anyone who consults it understands that they must not copy it or use material from it except as licenced, permitted by law or with the consent of the author or other copyright owners, as applicable

Access to this thesis/portfolio in print or electronically is restricted until (date).

Signed on behalf of the Doctoral College (also print name)

Declaration of any Previous Submission of the Work

The material presented here for examination for the award of a higher degree by research has / has not been incorporated into a submission for another degree.

Candidate's signature

Declaration of Authorship

I am the author of this thesis, and the work described therein was carried out by myself personally.

Candidate's signature

Abstract

Lower limb amputees struggle with an impaired gait that traditional prostheses' performance cannot entirely correct. The development of powered prosthetic devices aims to solve this by replacing the power generating muscles. Powered prostheses will only be successful if the device is effectively controlled.

A human gait involves multiple control modes for different activities; for a non-amputee, transitions between these modes are seamless and natural. In a powered prosthetic device, these levels of control need replicating. The correct selection of gait mode based on an individual's intent is crucial. Machine Learning (ML) methods are a promising avenue for this. However, their applicability to amputees is under-researched; this is partly due to the difficulty in collecting amputee gait data. This research aims to investigate ML methods for Locomotion Mode Recognition in amputees while reducing the data requirements for its implementation.

An extensive public data set of gait data is collected using novel wireless Inertial Measurement Units (IMU) and a companion smartphone app for labelling activities in real-time. The gait data set is used to investigate the performance of an Long Short Term Memory (LSTM) network for non-amputees. The analysis identifies that the model primarily classifies activity type based on data around early stance, a period with significant difference between individuals. The model also struggles to generalise to novel unseen users due to over fitting to the subjects' individual gait traits. Therefore personalisation is required.

Subsequently, methods for personalisation are investigated. Transfer learning is identified as a promising research field. However, its application to IMU amputee gait data has not previously been demonstrated. A novel method for dividing continuous, unstructured and poorly distributed gait data is developed to investigate personalisation methods: this successfully improves classification performance and reduces data requirements in both amputees and non-amputees.

Acknowledgements

I would like to thank:

Dr. Pejman Iravani for his support, guidance, and constant supply of exciting distractions.

Sarena Matheson for her endless love and belief through all the ups and downs. That and her not insignificant patience both reading and listening to me explaining my work.

I would also like to thank my friends and family who have provided constant support and encouragement throughout. Additional thanks go to the select few (Alex Powell, James Male, Michael Morris and Oliver Sherratt) who had the honour of proofreading.

Richard tucker for the loan of the sensors and providing me with experience and ideas to effectively use them

Finally, to Dr. Jawaad Bhatti and the staff at Blatchford for help collecting amputee data as well as assistance in understanding the issues and challenges that amputees face and the difficulties involved in developing powered prosthetic devices

Table of Contents

List of Figures	ix
List of Tables	xiii
List of Abbreviations	xv
1 Introduction	1
1.1 Motivation	1
1.2 Challenges, Control and Machine Learning	1
1.3 Hypothesis	2
1.4 Thesis Structure	3
1.5 Contributions	3
2 Background	7
2.1 Biomechanics of Gait	7
2.2 Lower Limb Prosthesis	13
2.3 Sensors	17
2.4 Machine Learning	22
2.5 Long Short Term Memory (LSTM)	25
2.6 Locomotion Mode Recognition	26
2.7 Conclusions	34
3 Methodology	37
3.1 Recording Hardware	37
3.2 Android Application Design	40
3.3 Collection Methodology	43
3.4 Summary of Data	47
3.5 Post-Processing	48

3.6	Machine Learning Methods	53
3.7	Discussion and Conclusions	55
4	Understanding LSTM Locomotion Mode Recognition Behaviour	59
4.1	Introduction and Commentary	59
4.2	Article: Understanding LSTM Network Behaviour of IMU-Based Locomotion Mode Recognition for Applications in Prostheses and Wearables	61
4.3	Post-commentary	89
5	LMR Model Personalisation	91
5.1	Methods and Materials	92
5.2	Baseline Model Performance	97
5.3	Data Supplementation	102
5.4	Transfer Learning	103
5.5	Discussion and Conclusions	109
6	Applicability to Amputee Data	113
6.1	Methods and Materials	114
6.2	Baseline Model Performance	115
6.3	Data supplementation	118
6.4	Transfer Learning	119
6.5	Discussion and Conclusions	122
7	Conclusions and Further Work	127
7.1	Conclusions	127
7.2	Further Work	129
A	Tables of Results	133
B	Ethics Forms	147
C	Sensor Datasheet	165
D	Sample Data	169
	References	185

List of Figures

2-1	Human Gait Cycle during level walking	8
2-2	Bio-mechanical planes of the body	9
2-3	Sagittal plane motions of the ankle	9
2-4	Stair ascent gait cycle	10
2-5	Stair descent gait cycle	11
2-6	Example prostheses types	14
2-7	Generalised control framework for active lower limb prostheses	15
2-8	Example state machine for mid level control of a lower limb prosthetic ankle	16
2-9	IMU sensor and example data	19
2-10	Force Myography sensor and example data	19
2-11	EMG sensor and example data	20
2-12	IMU sensor and example data	21
2-13	Target gait parameter locations for wearable sensing	22
2-14	LSTM with Forget Gate	26
2-15	Stair Ascent and Descent Heuristic	28
2-16	An illustration of sensor-based machine learning activity recognition	29
3-1	Movesense Wearable IMU	38
3-2	Movesense sensor attachment locations	39
3-3	Movesense Bluetooth Low Energy transmission packet structure	41
3-4	Android App Design Process	42
3-5	Data-logging android application user interface.	42
3-6	Data-logging Android App	43
3-7	Example of data recording environments	45
3-8	Comparison of right ankle angular velocity during ramp ascent	46
3-9	Comparison of right ankle acceleration during ramp ascent	46
3-10	Comparison of right ankle angular velocity during ramp descent	46
3-11	Comparison of right ankle acceleration during ramp descent	46

3-12	The hierarchical structure of the data recordings and terminology.	49
3-13	Flow Diagram of Sensor Data ETL process	49
3-14	Example of sensor clock drift correction.	51
3-15	Flow Diagram of the Machine Learning ETL process	52
3-16	Sliding window generation	53
4-1	Human Gait Cycle during level walking.	64
4-2	Gait events extracted from the sagittal plane gyroscope signal.	64
4-3	Unfolded Recurrent Network.	65
4-4	LSTM unit with input and output connections.	66
4-5	Subject wearing the Movesense IMU sensors	70
4-6	Custom Android app with connected sensors	71
4-7	Raw data input and pre-processing flow diagram.	72
4-8	Machine-Learning Model Architectures.	73
4-9	Division of the population to form the training, validation and test sets . .	74
4-10	Gait trends for the right shank x accelerometer and y gyroscope	77
4-11	Hidden state of single unit LSTM model with x axis accelerometer as its input. .	79
4-12	Hidden state of single unit LSTM model with x axis accelerometer as its input. .	80
4-13	Model accuracy for hyper-parameters of different LSTM hyper-parameter . .	82
4-14	Classification accuracy for training with different numbers of participants . .	83
4-15	128 timestep, 6 unit confusion matrices.	83
4-16	128 timestep, 32 unit confusion matrices.	83
4-17	Miss-classifications and labelled activity locations	84
4-18	128 \times 6 Transition Model.	85
4-19	128 \times 32 Transition Model.	85
5-1	Shank angular velocity during different activities for each target subject . .	94
5-2	Per-episode data division	95
5-3	Illustration of LSTM machine learning model architecture	97
5-4	LSTM accuracy for training with varying amount of target data	101
5-5	Fine-tuning an LSTM model using target data	108
6-1	Amputee and non-amputee shank angular velocity for different activities . .	116
6-2	Classification accuracy using increasing quantities of amputee target data . .	119
6-3	Accuracy of re-training a pre-trained model using amputee target data . . .	121
D-1	Example left ankle accelerometer data	173
D-2	Example left ankle gyroscope data	174
D-3	Example right ankle accelerometer data	175

D-4	Example right ankle gyroscope data	176
D-5	Example left hip accelerometer data	177
D-6	Example left hip gyroscope data	178
D-7	Example right hip accelerometer data	179
D-8	Example right hip gyroscope data	180
D-9	Example chest accelerometer data	181
D-10	Example chest gyroscope data	182

List of Tables

3-1	Compression of sensor readings, scaling factors and resultant accuracies . . .	40
3-2	Summary of Data collected	48
3-3	Summary of Test Subject Demographics	48
3-4	Data specification	48
4-1	Statement of Authorship	60
4-2	Quantity of data collected for each activity.	71
4-3	Summary of simplified model performance.	78
5-1	Table of quantities of data samples and episodes collected for each target subject.	92
5-2	Personalisation Target Subject Demographics	93
5-3	Confusion matrices of a general model presented with target subject test data	99
5-4	Confusion matrices for a bespoke non-amputee LMR model presented with target subject test data	100
5-5	Table of classification accuracy for Subject 01 for a model trained using varying amounts of Source and Target training data	104
5-6	Table of classification accuracy for Subject 03 for a model trained using varying amounts of Source and Target training data	105
5-7	Table of classification accuracy for Subject 09 for a model trained using varying amounts of Source and Target training data	106
6-1	Summary of amputee gait data collected	114
6-2	Confusion matrix of general models presented with target subject test data	117
6-3	confusion matrix for a bespoke amputee LMR model presented test data . .	118
6-4	Table of classification accuracy for amputee test data for a model trained using varying amounts of Source and Target training data	120

A-1	Table of results for classification performance of different size LSTM networks trained with varying amount of target subject data for subject 1.	134
A-2	Table of results for classification performance of different size LSTM networks trained with varying amount of target subject data for subject 3.	135
A-3	Table of results for classification performance of different size LSTM networks trained with varying amount of target subject data for subject 9.	136
A-4	Classification accuracy for a 32 unit LSTM model retrained with increasing amount of target data	137
A-5	Classification accuracy for a 32 unit LSTM model with the dense layer retrained with increasing amount of target data	139
A-6	Classification accuracy for a 32 unit LSTM model with the LSTM layer retrained with increasing amount of target data	140
A-7	Table of results for classification performance of 32 unit LSTM networks trained with varying amount of amputee subject data	142
A-8	Table of results for classification performance of 32 unit LSTM networks personalised to an amputee using transfer learning	143
A-9	Table of results for classification performance of 32 unit LSTM networks with the dense layer personalised to an amputee using transfer learning.	144
A-10	Table of results for classification performance of 32 unit LSTM networks with the LSTM layer personalised to an amputee using transfer learning.	145
D-1	Table of demographic data for study subjects	170
D-2	Data samples of non-amputee data collected during the first phase of collection	171
D-3	Data samples of non-amputee data collected during the second phase of collection	171
D-4	Data samples of first hand amputee data collected during the third phase of collection	171

List of Abbreviations

ADL	Activities of Daily Living	ML	Machine Learning
ANN	Artificial Neural Network	MLP	Multi-Layer Perceptron
BLE	Bluetooth Low Energy	RA	Ramp Ascent
CNN	Convolved Neural Network	RD	Ramp Descent
COM	Center Of Mass	ReLU	Rectified Linear Unit
COTS	Commercial Off The Shelf	RNN	Recursive Neural Network
CSV	Comma Separated Values	SA	Stair Ascent
DPS	Degrees per Second	SD	Stair Descent
EMG	ElectroMyoGraphy	SDK	Software Development Kit
ESR	Energy Storage and Return	SVM	Support Vector Machine
ETL	Extract Transform Load	TO	Toe Off
FID	Frechet Inception Distance	YAML	YAML Ain't Markup Language
FMG	Force Myography		
GATT	Generic Attribute Profile		
GRF	Ground Reaction Force		
GRU	Gated Recurrent Unit		
HAR	Human Activity Recognition		
HS	Heel Strike		
IC	Initial Contact		
IMU	Inertial Measurement Unit		
LMR	Locomotion Mode Recognition		
LOOXV	Leave One Out Cross Validation		
LSTM	Long Short Term Memory		
MARG	Magnetic Angular Rate and Gravity		

Chapter 1

Introduction

1.1 Motivation

Lower limb amputation affects a small but significant portion of the population. Predictions, however, suggest that this number will continue to rise. More than one million amputations occur globally, that is one every 30 seconds.[1] Limb loss often occurs due to traumatic injuries, certain diseases, and forced amputation due to surgery, increasingly resulting from vascular disease or diabetes[2, 3].

Regardless of amputation cause, lower limb amputees require more energy to walk than their non-amputee peers[4]; 10-40% for trans-tibial[5, 6] and greater than 70% for trans-femoral[7]. The loss of a lower limb dramatically impedes movement[8, 9, 10], reducing amputees' quality of life and increasing the risk of further compensatory injuries. A powered prosthesis could effectively replace the lost limb, reducing energy expenditure during walking and rebalancing gait[11]. To effectively control the prosthetic requires knowledge of the user's intended activity. This knowledge must be obtained in real-time only using information gathered through local sensors.[12]

1.2 Challenges, Control and Machine Learning

For the non-amputees, transitioning between different locomotion modes is seamless as both legs adapt to the activity without thought. For a leg prosthetic to feel truly natural, its controller must be capable of the same seamless behaviour. The identification of appropriate locomotion mode is a vital component of this.

Variations in gait between individuals are substantial enough that it is possible to identify an individual based solely on their gait[13, 14]. For amputees, inter-subject differences in gait are more substantial. On top of the normal gait variations the level of amputation and any compensatory mechanisms can have large effects. Therefore the individual tuning or personalisation of prosthetic controllers to an individual is of key importance.

Additional complexities come from the need to adapt to different environments. A change in environment, such as moving from a paved path to a woodland trail will result in changes in sensor signals. This is especially challenging to deal with as it will never be possible to collect sample data for all environments that a prosthetic device may operate in.

Accounting for both these challenges requires the need for a highly tune-able controller. Machine Learning (ML) has been shown to be effective at extracting information for new environments as well as learning behaviour unique to an individual. The problem with ML is that it requires a large amount of data in order to achieve a high enough performance to not compromise the safety of the device. Collecting large amounts of gait data from an amputee is challenging given their reduced mobility.

1.3 Hypothesis

This thesis explores a hypothesis in the cross-cutting domain of human gait, control of prosthetic device and machine learning approaches.

The hypothesis is:

A Machine Learning approach based on Long Short Term Memory (LSTM) architecture can be used to predict gait modes with data requirements reduced through a transfer learning approach.

This hypothesis can be explored further by:

- Collection of a large gait data set to experiment with LSTM ML methods for Locomotion Mode Recognition (LMR).
- Improving understanding of the underlying mechanisms for how a LSTM network classifies gait and the potential limitations of this.
- Development of ML schemes for individual personalisation of machine learning models to reduce training data requirements.

1.4 Thesis Structure

The structure of each chapter is as follows.

- In Chapter 2, an overview of the background around machine learning and gait are presented.
- Chapter 3 presents the methodology used for data collection and Machine Learning.
- In Chapter 4 the Journal article “Understanding LSTM Network Behaviour of IMU-Based Locomotion Mode Recognition for Applications in Prostheses and Wearables” is presented.
- Chapter 5 presents a method for evaluating personalisation LMR models from a set of real-world continuous gait data. The Chapter also demonstrates a simple personalisation method to improve classification performance over realistic baselines.
- Chapter 6 applies the methods developed in the previous chapter to first hand trans-tibial amputee data.
- Finally Chapter 7 present Conclusions of the thesis and suggestions for future work.

1.5 Contributions

The main contribution of this work is the demonstration of a practical transfer learning approach to producing an activity recognition system for an amputee. The specific contributions are as follows.

- Presentation of the current state of the art in ML methods for locomotion mode classification from amputees gait sensor data.
- Development of a Android Application and wireless sensor system for collection of a unsupervised system for the large scale collection of labelled gait data is developed.
- Collection of a new publicly available data set of unsupervised gait data collected in a natural environment.
- Contributes to a better understanding of Long Short Term Memory for LMR networks.
- Demonstrates the need for personalisation of ML models to achieve $> 80\%$ model accuracy.

- Creation of methods for evaluating personalisation LMR models from a set of real world continuous gait data
- Demonstration of personalisation methods for improving individual locomotive mode classification accuracy.
- Demonstration of high applicability of personalisation methods amputee gait classification.

Chapter 2

Background

This chapter presents the background and literature on locomotion mode recognition systems for prostheses. The chapter contains seven sections. Firstly background around human gait and prostheses are presented in Sections 2.1 and 2.2. Following this, Section 2.3 reviews wearable sensors used for gait analysis. Sections 2.4 and 2.6 contain an introduction to machine learning and its applications for Locomotion Mode Recognition. Finally, concluding remarks about areas of research interest are presented in Section 2.7.

2.1 Biomechanics of Gait

Human locomotion involves the smooth advancement of the body through space with the least mechanical and physiological energy expenditure. While the primary goal of walking is forward motion, the mechanism that achieves it is based on the need to maintain a symmetrical and low amplitude displacement of the body's COM in both vertical and lateral directions. Low displacement conserves both kinetic and potential energy and is the principle of biological conservation of energy.[15]

Gait is a highly individualistic personal trait with many factors that affecting it[16]. A performant gait is a coherent highly energy-efficient mechanism for forward propulsion of the body. It naturally adapted to different environmental conditions and disturbances to achieve high level of stability throughout the gait cycle[17, 18]. With regards to lower-limb prosthetic the mechanics of locomotion are of most interest. Within this section the terminology that will be used to with reference to the human gait and the high level biomechanics of locomotion are presented.

Different ambulation modes require fundamentally different control sequences for operating powered prosthetic limbs[19]. There are a large number of possible locomotive activities depending on the environment and movement speed. In the study of activity recognition level-ground walking, ramp ascent, ramp descent, stair ascent, and stair descent are the most commonly studied[20, 21, 22, 23, 24, 25, 26, 10, 27] and cover the most basic locomotion activities used for forward motion[28]. Labarrière et al. identified these as the most commonly investigated as they require no equipment or skill to perform[26].

2.1.1 Gait Terminology

A complete gait cycle is the basic unit of gait analysis. A cycle, by convention, begins when one foot comes into contact with the ground, Initial Contact (IC), and is complete when the same foot contacts the ground again. Another name for IC is Heel Strike (HS), as the heel is the most common initial point of contact. Conversely, the point at which the foot leaves the ground is Toe Off (TO). The name arises as the toe is always the last point of contact with the ground.[29, 17]

The gait cycle can be further sub-divided into two phases. The distinct phases, stance and swing, are physically indicated by the foot’s contact with the ground. Stance occurs when the foot is in contact with the ground, and swing occurs when the foot is off the ground. HS marks the transition from swing to stance and TO stance to swing. When considering both limbs, there are additional descriptors. These include single support when only one foot is in contact with the ground and double support when both feet are in contact with the ground. Figure 2-1 illustrates a complete gait cycle and the key events.[29, 17]

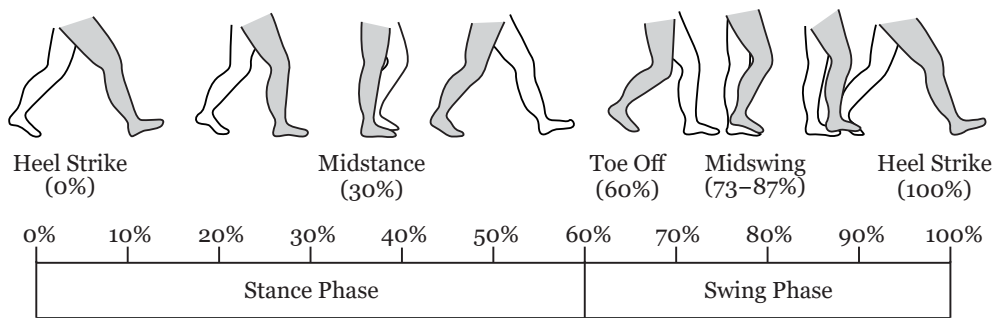


Figure 2-1: Human Gait Cycle during level walking. The percentage timings of the gait events are approximate; they vary depending on the individual and environment.[30]

Movements of the human body mainly occur in three planes: sagittal, frontal/mediolateral and horizontal/transverse. The plane’s intersections occur either at a joint centre or the body’s Center Of Mass (COM). The sagittal plane is the vertical plane passing from the

rear (posterior) to the front (anterior), dividing the body left and right. The mediolateral plane passes from left to right, dividing the body into posterior and anterior halves. The transverse plane divides the body into the top (superior) and bottom (inferior) halves.[31] Figure 2-2 shows an illustration of the three planes

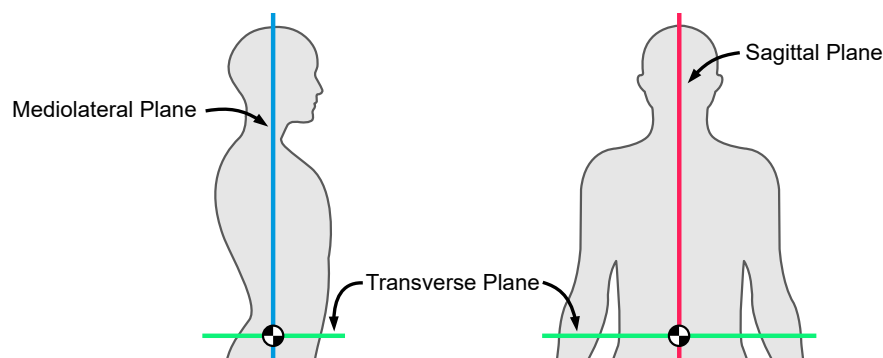


Figure 2-2: Bio-mechanical planes of the body. The sagittal plane is the vertical plane dividing left and right body halves. The mediolateral plane divides the body into front and rear halves. Finally, the transverse plane divides the body into the top and bottom halves.

The primary movement of the ankle occurs in the sagittal plane; these are the raising and lowering of the foot. The two motions are plantar-flexion, moving the foot downwards, and dorsiflexion, lifting the foot upwards.[31] Figure 2-3 show a visual of the ankle movement. Plantar-flexion happens towards the end of the stance phase when the foot pushes off the ground. Dorsiflexion occurs during the early swing phase to provide enough toe clearance as the foot passes under the body.[32]

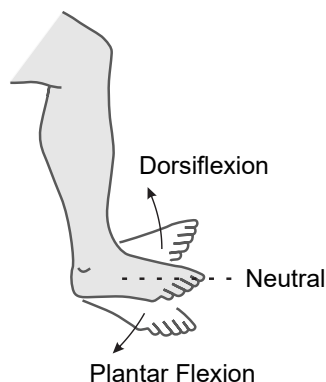


Figure 2-3: Sagittal plane motions of the ankle. Plantar flexion is the lowering of the foot; dorsiflexion is the raising of the foot.

There are many different metrics for quantifying gait. These vary from easily measurable values such as step rates and distances to more involved metrics such as energy expenditure and efficiency.[33, 34]. Measurements of energy expenditure are not possible directly.

Energy expenditure must be is calculated indirectly, this is commonly done through measurement of the volume of oxygen consumed and carbon-dioxide produced. The equipment required for measurements restrict energy measurements to a lab environment.[15]

2.1.2 Variation with Locomotive Activity

The previous section described the pattern of gait that occurs during a level walking locomotion. The human gait cycle can efficiently adapt to different terrain and obstacles. Changes in the locomotive activity require a change in gait actions to accomplish the movement[19]. Additional muscle actions are required to raise and lower the COM during these actions[35].

This section presents bio mechanical difference between five different locomotive movements, Walking, Stair Ascent (SA), Stair Descent (SD), Ramp Ascent (RA), Ramp Descent (RD). Ramps are considered any surface with a slope sufficiently steep so that a change in locomotive action is required. For each activity, there is a description of the differences in human gait compared to walking.

2.1.2.1 Stair Ascent

During SA, the COM must move upwards, requiring net positive work; this requires more significant muscle activity. SA can be divided into three phases: weight acceptance, pull-up and forward continuation. The knee dominates during weight acceptance and pull-up with support from the hip and ankle. While, during forward continuation, the ankle generates a large amount of energy resulting in an upwards motion of the COM. The ankle angle differs from a horizontal walk, mainly at the late swing phase and early stance. While lifting the foot to the next tread, the edge is avoided by a small dorsiflexion and moving the knee backwards.[36] The changes in gait cycle are illustrated in Figure 2-4.

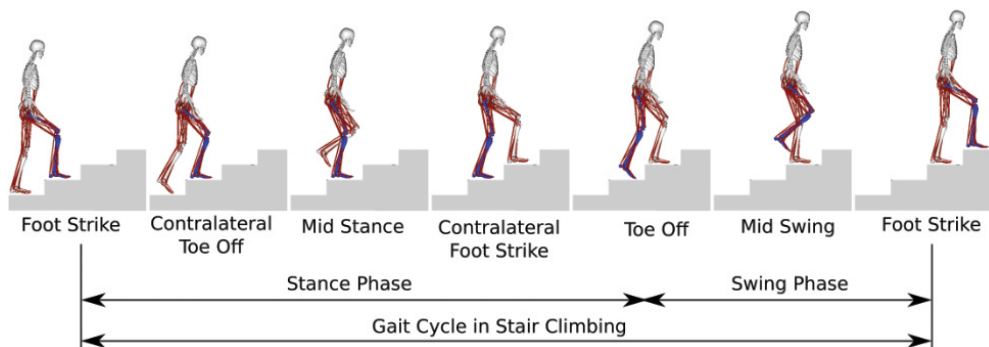


Figure 2-4: Stair ascent gait cycle[37]

2.1.2.2 Stair Descent

During SD, the ankle angle differs from horizontal in the swing phase when moving the limbs down. The change in ankle angle is most notable as a dorsiflexion action to reaching the toe downwards. The change in ankle angle results in the toe being the point of IC. Energy is transferred from the ankle into the knee at IC. Due to the downwards, COM motion reflects in a smaller force at push-off. There is also less muscle activity for vertical movements due to the smaller stride length.[36] The changes in gait cycle are illustrated in Figure 2-5.

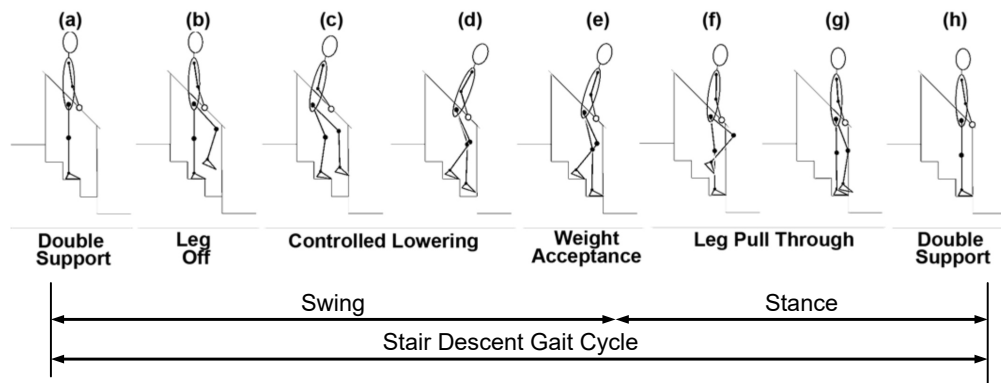


Figure 2-5: Stair descent gait cycle[38]

2.1.2.3 Ramp Ascent

As with SA, RA requires additional energy expenditure to move the COM upwards[35]. Walking uphill can take three times as much energy as walking on flat ground[39]. Gait parameters also vary with the slope of the surface[40]. Knee flexion and ankle dorsiflexion increases at heel strike as the foot aligns with the surface. These changes in gait require an increased range of motion and additional muscle power generation.[41]

2.1.2.4 Ramp Descent

For moderate slopes, RD is similar to level walking. However, the lowering of the COM requires additional energy to be absorbed[35]. Walking downhill takes only half as much energy as walking on the level ground[39]. As with RA, the gait adjusts to the slope.

The above describes the steady-state motions during typical locomotive activities. The human gait can also smoothly transition between different locomotive modes and handle perturbations and disturbances.[42]

2.1.3 Gait variations between Amputee and Non-Amputee

Amputees suffer from a wide range of gait abnormalities when compared to non-amputees. These occur both due to mechanical constraints of a prosthesis and compensatory actions due to the functional loss of muscles [43, 44]. These adaptations are individual and unique to the amputee and their particular amputation [45, 46]. General trends in the adaptation may be drawn, such as asymmetrical gait, slower walking speed, and higher energy demands[47].

Amputee gait is significantly more asymmetrical compared to that of a non-amputee, with amputees relying more heavily on their intact limb [48, 49]. Asymmetry is seen in many gait features, including stance and swing periods and hip, knee and ankle joint moments[50, 51]. A common explanation for this is the reduced push-off capacity of the prosthetic leg, but it may also be due to discomfort of the prosthetic and other compensatory mechanisms[52].

Energy expenditure measurement have proven to be a reliable method of quantitatively assessing the penalties imposed by gait disability [15]. Studies have shown that trans-tibial amputees using passive prosthesis use 10-40% more energy to walk at the same speed when compared to their non-amputee peers [5, 6], with trans-femoral amputees requiring more than 70% [7]. This additional energy expenditure dramatically impedes ambulation. Amputees are half as active as their non-amputee peers and prefer 30-40% slower walking speeds [53, 54, 11, 55]. The loss of significant power generating muscles after amputation also leads to a marked asymmetry in gait between limbs [56].

Amputee subjects walk at a 29% lower comfortable speed than a non-amputee [57, 58]. They also walk with a larger stride width than normal subjects [57]. To achieve higher walking speeds amputees prefer to adjust their stride length rather than with their step rate [57, 43].

Insufficient mid swing toe clearance of a prosthetic foot is a well-recognised inadequacy for lower limb prosthesis user due to inability to adequately dorsi-flex the ankle. The lack of toe clearance results in an increased risk of tripping [45]. Compensatory mechanisms are employed by the amputee to reduce tripping risk. A common mechanism is elevating the prosthetic hip during mid-swing referred to as Vaulting or hip-hiking. Elevation is achieved by employing an early heel rise of the intact limb.[59, 45]. Hip-hiking increases the COM height during swing on the amputated limb when compared to a non-amputee. The increased COM reduces gait efficiency and the asymmetry of the gait can lead to injury.

Lower limb amputees spend more time in stance on their intact limb and less on their prosthetic limb [60]. The double-limb-support phases of the intact and prosthetic side are unequal in most amputee subjects [57]. In particular the different characteristics of the

prosthesis relative to the sound limb results in a longer swing time for the prosthesis [47]. This asymmetry increased with walking speed [60].

An amputee also have an asymmetric Ground Reaction Force (GRF). Able-bodied subjects have less than 10% force asymmetry during walking while unilateral amputees have up to 23% force asymmetry depending on the type of prosthesis used [60]. There is also a decreased GRF on prosthesis compared to a non-amputee[46]. Many lower limb amputees attempt to maximise the capacities of their intact limb to counteract the limitations of their prosthetic device explaining some of the GRF asymmetry[61].

2.2 Lower Limb Prosthesis

A lower-limb amputation involves the removal of part, or multiple parts, of the lower limb. The practice of amputation for injury or disease is centuries old and likely first occurred in pre-historic times[62]. The level of amputation can be either minor or major, depending on the amputation site. Major lower limb amputations occur above the ankle. The two most common major lower limb amputations are trans-femoral (above the knee) and trans-tibial(below the knee).

An artificial or prosthetic limb can replace an amputated extremity. The prosthesis aims to restore natural behaviour by emulating/augmenting the action of a missing limb.[12] The ability of a prosthesis to mimic the function and appearance of the lost limb can vary significantly.

The earliest known prostheses are recorded in Indian literature between 1500 to 800 b.c. although artificial limbs are probably much older than that[63]. Over the years, prostheses have gone from barely functional to sophisticated devices aiming to replicate lost functionality. Artificial limb design, fitting and manufacture has advanced considerably within the past 50 years and is now a burgeoning research field.[62]

The current primary technologies for prosthetic legs are: Energy Storage and Return (ESR), hydraulics, semi-active and active devices[1]. Figure 2-6 shows examples for each type.

Each technology offers varying levels of assistance, with amputee suitability and preference informing selection. The most common form of prostheses is the ESR foot. During walking, an ESR prosthesis reduces energy expenditure by storing energy through spring compression during heel loading, releasing it during late stance to push-off.[1]

Passive devices such as an ESR cannot provide the net positive mechanical power needed during many activities of daily living, such as ascending stairs or standing up from a seated

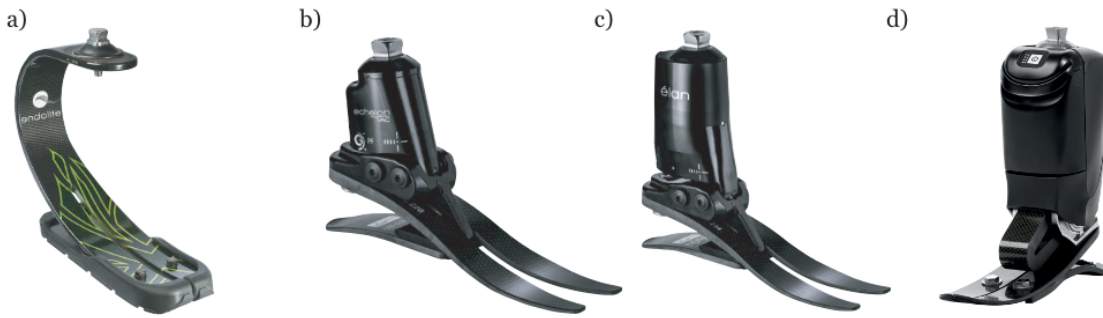


Figure 2-6: Example prostheses types: a) ESR (Blatchford BladeXT), b) Hydraulic (Blatchford EchelonVAC), c) Semi-active (Blatchford Elan), d) Active (Ottobock's biOM) (a-c)[64] (d)[65]

position[22]. To fully restore gait functionality, the prostheses must replace the lost power generating function of the amputated limb.

A powered active prosthetic device could provide the full power-output capabilities of the corresponding physiological joints. It could thus enable gait patterns resembling those of unaffected persons across various activities and terrain[12, 66]. According to Au et al., The use of an active prosthetic can reduce metabolic energy usage by 14% in trans-tibial amputees, even with a two-fold increase in prosthetic weight[55].

Ottobock's BiOM EmPOWER active prosthetic ankle is currently the only active prosthetic device on the market.[67] The device is a spin-out from research undertaken by MIT in 2011.[65] Analysis of the EmPOWER ankle has shown a reduction in the metabolic cost of walking by 8% and an increased walking velocity of 23%[6]. Össur produces the Proprio Foot, a semi-active ankle that lifts the toe to increase ground clearance. The Proprio foot has also been integrated with their Rheo Knee to form a coordinated leg for trans-femoral amputees capable of stair descent.[68]

2.2.1 Control Requirements for Powered Prosthesis

The human body represents a well-balanced walking machine that performs periodic, stable, and energy-efficient gait through highly sophisticated mechanics and control, which are not easy to replicate[18]. The human nervous system uses different control strategies to adapt to individual locomotive activities[69, 22]. These adaptations occur automatically in a healthy gait cycle. A prosthetic controller must implement multiple control modes to replicate this lost functionality.

An active prostheses controller must implement several concurrent processes to control pros-

theses effectively. Tucker et al. present a generalised hierarchical framework for structuring a prostheses controller. Hardware control forms the lowest level. The top-level controller implements a perception system to determine user intent or activity mode. Therefore, an accurate perception of the users' intended action is paramount[1, 70]. An intermediate level controller performs translation of intent to hardware state demands. An illustration of the hierarchical controller is provided in figure 2-7.[12]

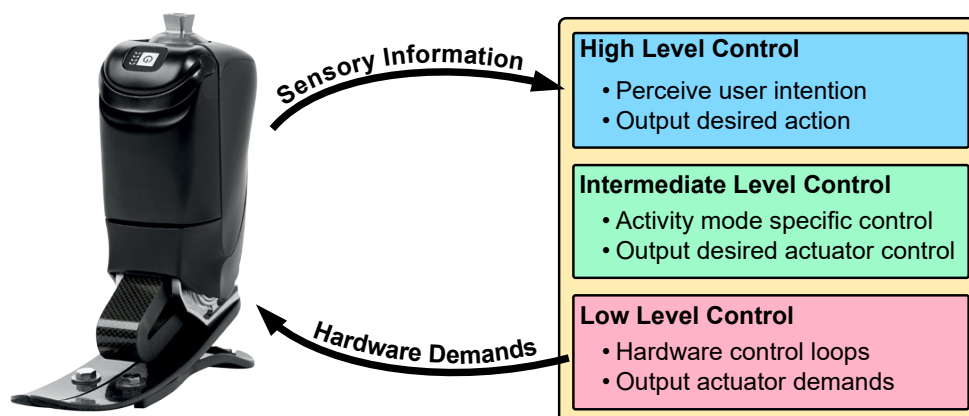


Figure 2-7: Generalised control framework for active lower limb prostheses

2.2.1.1 High Level Controller

As different activities require changes to gait pattern multiple mid-level controllers are required for each locomotive tasks; for example walking, standing still or stair descent. Selection of the appropriate mode is critical to performance of this controller as selection of the wrong mode will result in undesirable behaviour. The high level controller is responsible for the selection of the appropriate mid-level controller.[12]

There are many different forms of mode selection including manual user entry, heuristic threshold methods[71, 72, 73] and machine learning methods[74, 75]. All of these methods use sensors to interpret user intent and environment around them to make an appropriate mode selection.

2.2.1.2 Mid Level Controller

The mid controller is responsible for determining the demand physical state of the device throughout the gait cycle. The selection of demand state is driven by the which phase of the gait cycle the user is in, and the current activity mode. The demand states have the form of a position/velocity, torque or, impedance of different prosthetic components which

are fed to the low level controller to enforce.[12]

Most mid-level controllers have a finite number of modes corresponding to a series of different activity sequences. Researchers are also investigating continuous or mode free controller to reduce the need for bespoke state machines per activity.[76, 77]

Accurate determination of the current gait phase is critical to outputting appropriate demand signals. When done correctly this allows the prosthetic to provide a natural walking gait and to make the most use of any power generating/absorbing capacity For example Yu et al. presented a prosthetic ankle that could provided powered plantar-flexion. From the work it was identified that accuracy timing of the plantar-flexion was critical. Plantar-flexing too early resulted in the ankle lifting the body upward instead of pushing the body forward. Powering the plantar-flexion too late resulted in a lack of support of the body weight, with the amputee in danger of stumbling.[78]

Sup et al. illustrated an example mid-level state machine as shown in Figure 2-8. The selection of modes is achieved based on threshold for different sensor inputs being reached. For example mode one begins after mode four when the measure ankle loading exceeds a manually set threshold. The state machine moves to mode two when the ankle angle exceed another threshold value.[79]

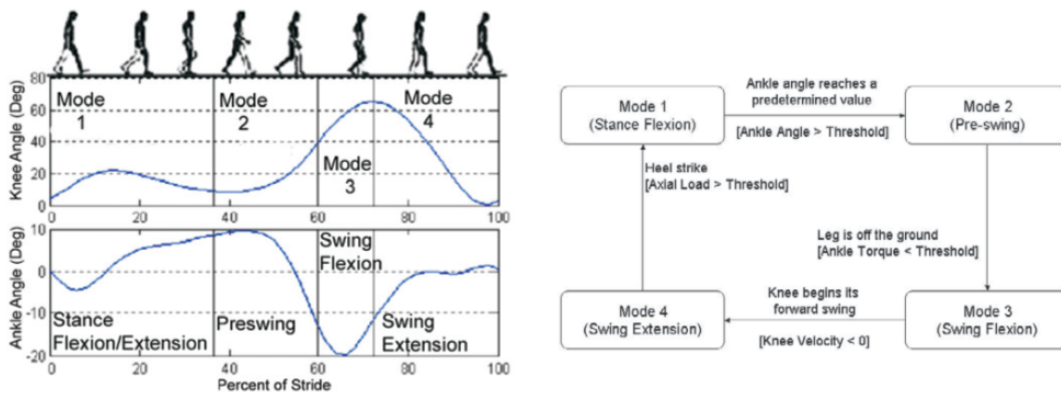


Figure 2-8: Example state machine for mid level control of a lower limb prosthetic ankle [79]

2.2.1.3 Low Level Controller

The desired device state is passed to the low-level controller. The low-level controller then used closed loop feedback to actuate the prosthetic device. Feedback is based on the error between the current and desired state. There are many forms of controller that are used to achieved this including feed-forward or feedback control.[12] Gao et al. used

a Proportional-Derivative controller to regulate the motor current and provide closed loop torque control[80]. Yu et al. used a proportional-integral based closed loop velocity controller to provided velocity control for a hydraulic pump. This was used to regulate the hydraulic pump pressure to drive the hardness and position of a hydraulic actuator.[78]

2.3 Sensors

To perceive a person’s intent, we must take measurements of them[1, 70]. Different sensors allow for different measurements. Each measurement can be of physical quantities about the person or their surrounding environment. Appropriate selection of sensors is therefore of critical importance. Criteria for selection include the type, richness of information and impact on the user.[12]

The impact of a sensor considers its invasiveness both physically and in privacy. The physical impact must consider how the discomfort of a sensor affects the performance of an action. For example, a sensor of fewer than 300 grams mounted to a shoe does not affect gait significantly[81]. In contrast, many wires rubbing on the leg may. There are also practical concerns, such as ease of use.

Sensing systems divide into three categories: neural, mechanically intrinsic or environmental signals. Neural sensors measure physiological electrical signals, such as brain activity or muscle activity. Mechanically sensors measure effects intrinsic to the device itself, such as acceleration. Environmental sensors measure the properties of the world around them, such as ambient light or pressure.[82, 12]

Recent trends in sensing technology have been towards wearable sensors. The demands of the modern smartphone have primarily driven this development. New smaller and more precise sensors have opened up the possibility for making previously lab bound measurements in a more natural environment. Smartphone sensing technology is highly applicable and widely used in active prostheses[1]; this state of the art review will focus only on wearable technologies.

2.3.1 Types of Sensor

Acceleration and angular velocity are the most commonly mechanically intrinsic signals measured. A wide variety of wearable sensors can collect this information.[83, 12] Acceleration and angular velocity are the most commonly mechanically intrinsic signals measured. A wide variety of wearable sensors can collect this information. Other common wearable sensors include goniometer and inclinometers for angular displacements, and pressure

transducers or Force Sensitive Resistors (FSR) for foot loading and initial/terminal contact points. Ground Reaction Force (GRF) can also be measured using pressure transducers such as an FSR[84].

The most common wearable sensor used is the Inertial Measurement Unit (IMU)[83]. An IMU is a single integrated circuit containing a three-axis accelerometer and a three-axis gyroscope. The development of smartphones has dramatically reduced the sensors' cost, precision, and size. When an IMU includes a three-axis magnetometer, the sensor becomes a Magnetic Angular Rate and Gravity (MARG) sensor. Using one or multiple sensors allows the determination of limb or joint orientation, angular velocities and accelerations. Figure 2-9 shows a typical IMU sensor and sensor data received from it during different activities from a handheld IMU.

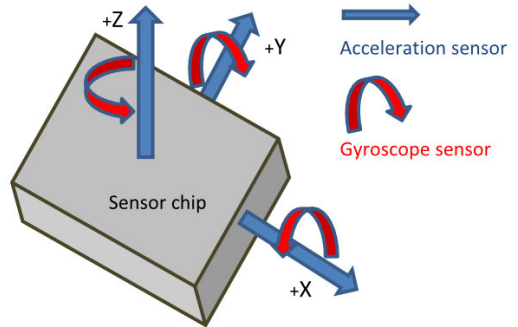
IMUs are widely available, low-cost, and easy to use while giving relatively high accuracy low latency measurements [85, 86]. As they measure physical movement they can be mounted in any location with no dependency on anatomical feature or proximity allowing for very low intrusion measurements[86]. Measurements of an IMUs have physical significance so can be directly interpreted[23]. IMUs also have low frequency requirements as they measure physical changes of the relatively low frequency human gait cycle. From literature most authors used frequencies around 50Hz[87, 88, 89, 90, 91].

However, due to the fact that the IMU captures the motion of the forearm, rather than the muscle signal, IMUs sense actions later than neural sensors[86, 23]. IMUs are also sensitive to alignment and placement accuracy. Misalignment results in a change in the axis over which an effect is measured while misplacement can change the magnitude of accelerations measured[92]. An IMU also suffers from long term drift and temperature dependent biases[93, 92, 85].

Force Myography sense volumetric and hardness changes in limbs caused by muscle contraction. Measurements are made by force-sensitive resistors or piezoelectric sensors wrapped around the circumference of the limb. Figure 2-10 shows an example thigh worn FMG sensor and typical sensor data that could be received from it.[23, 97]

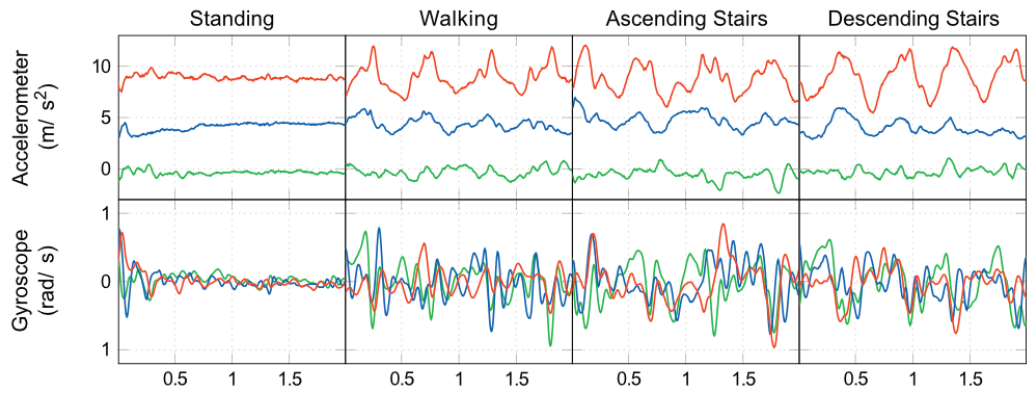
A potential advantage of FMG over neural measurements is that the muscle force estimated through FMG is less sensitive to fatigue. A substantial downside to this approach is a high sensitivity to motion artefacts, which may be significant given the nature of the physical attachment to the user.[12]

Neural signals provide a more natural interface for controlling prosthesis[99] but pose significant challenges in their use. Neural sensors detect muscle control signals, where as



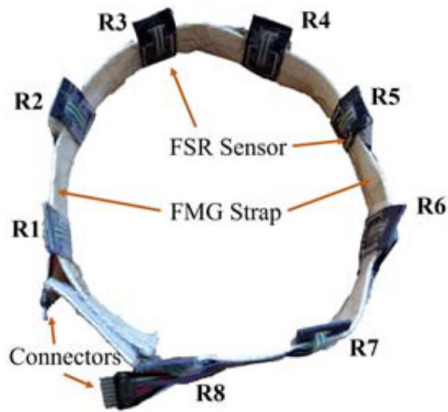
(a) Inertial Measurement Unit Sensor[94]

(b) Inertial Measurement Unit Axis[95]

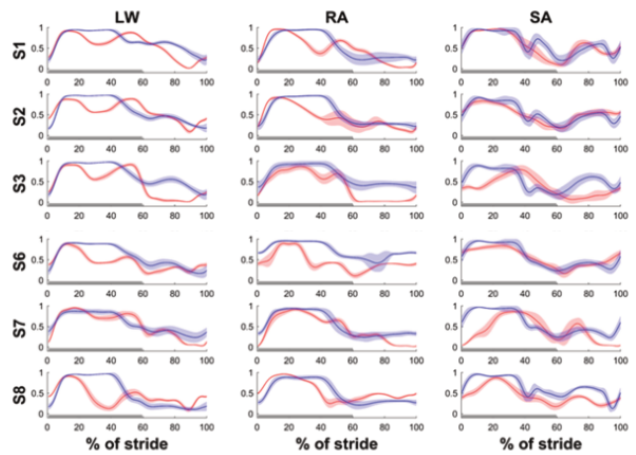


(c) Example IMU data[96]

Figure 2-9: IMU sensor and example data



(a) Force Myography Sensor[98]



(b) Example data[23]

Figure 2-10: Force Myography sensor and example data

mechanically intrinsic sensors measure the effect of muscle output. Therefore, neural signals can be detected in order of tens of milliseconds earlier[12], giving longer to perform classification. However, their output is often challenging to interpret due to low signal to noise ratio. Mechanical signals are often more straightforward to measure due to their lower impact and greater tolerance to placement and conditions.[82] ElectroMyoGraphy is the most common neural sensor. Electric potential produced by skeletal muscle activity is measured through electrodes attached to the muscle of interest. Figure 2-11 shows an example EMG sensors attached to a trans-tibial amputee’s stump and typical data that would be received from it.

EMG sensors give very early indication of muscle movement as they measure the signals driving muscle movement[23, 86, 100] combined with a very low latency measurement system they can provide a head-start over mechanical sensors[12].

However EMG sensors are sensitive to anatomical locations and require skill to fit correctly[101, 102]. They are susceptible to external factors, such as humidity and movement artefacts[86] and are potentially uncomfortable to wear directly on the skin[23]. EMG sensors also require high frequency sampling, exceeding 1000Hz, to detect signals sent to the muscle[100, 21]. As the human gait cycle occurs at 0.5 to 1.3Hz[103] this means sensor data must be captured at orders of magnitude above the gait cycle. This high frequency requirement increasing the complexity of capturing and analysing EMG data.

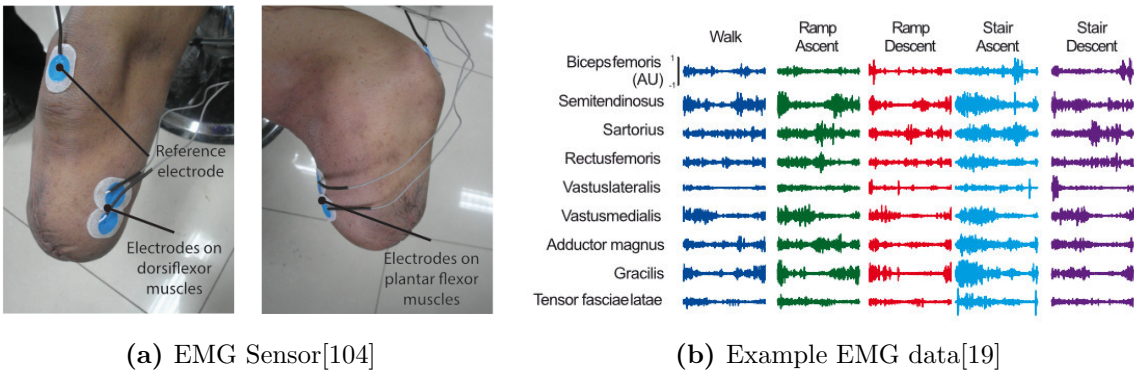


Figure 2-11: EMG sensor and example data

The environment around the user can provide context to local sensor data. Different environments increase the likelihood of encountering a particular terrain feature[12, 105]. Care must be taken using environmental sensors as they are highly noisy and susceptible to long term drift. Jin et al. investigated a evaluated three environmental sensors temperature, humidity, and ambient light. Jin measured repeatable changes in sensor data during different activities. For example during forward motion there was a temperature drop due to airflow

over the sensor package.[106] Figure 2-12 shows a typical temperature sensing integrated circuit and recorded data from a temperature. Fu et al. investigated the use of a barometric pressure sensor allowing for changes in altitude to be detected[107].

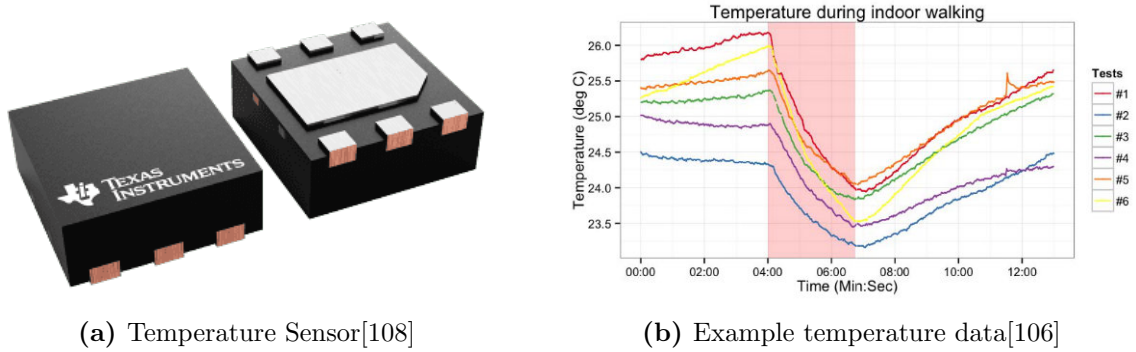


Figure 2-12: IMU sensor and example data

Data fusion is research field interested in integrating data from multiple sensors to achieve improved performance[100]. Sensors can either be multiple of the same sensor[109] or sensors of different modalities[100]. Data fusion is regarded as superior to methods that take inputs from a single data source with results indicating fusion methods can increase both accuracy and robustness[100, 110].

Chung et al. used eight IMUs placed across the body. By combining data from multiple sensors performance was improved. Chung also identified that the addition of multiple sensors meant sampling rates as low as 10Hz could be used. [109]

Liu et al. and Hu et al. both combined EMG and IMU sensors. Both found that fused IMU and EMG data gave higher accuracy than with just IMU data.[111, 110] Huang et al. fused EMG signals from the thigh with GRF measured from the pylon of a prosthetic devices. The results showed that the fused data outperformed methods that used only EMG signals or mechanical information alone.[100]

2.3.2 Sensor Placement

The placement of sensors is an important consideration; the chosen location should maximise the richness of data while minimising invasiveness[12].

The prosthetic device is the obvious mounting location providing a rigid and stable sensor platform for prosthetic users. Attachment is more challenging for biological limbs; temporary attachment of sensors by medical tape or elastic strapping is common. Consideration for changing the shape of the limb during movement is required.

Where suitable muscles are present at the skin, electrodes can be attached to the skin's surface above the muscle. Surface EMG presents the least invasive technique for the neural sensory system. However, they must be attached securely to the body to prevent artefacts from corrupting sensor movement readings and require individual calibration.

Shull et al. reviewed the location of wearable sensors used for gait analysis. Most articles used sensors that are sensitive in the mediolateral axis, with the knee, trunk and shank the most popular sensing location. Figure 2-13 shows a visual representation of Shull's findings.

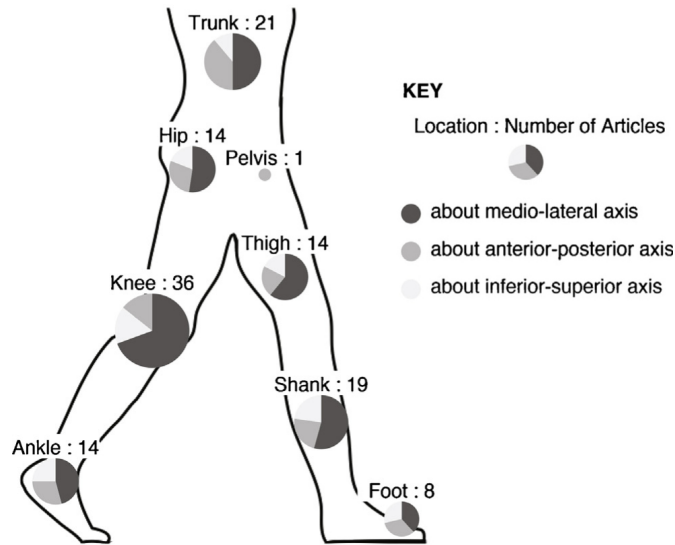


Figure 2-13: Target gait parameter locations for wearable sensing. The number of articles reporting gait parameter locations is indicated at each respective location. The diameter of each pie chart is proportional to the number of published articles reporting that gait parameter location. The relative proportion of kinematic and kinetic parameters about each of the three anatomical axes is indicated in the pie charts.

EMG sensors can be placed in multiple locations however for Locomotion Mode Recognition the calf and thigh are the most common locations[112, 113, 114]. There is less consistency for IMUs/ Shin et al. and Li et al. both placed sensors on the foot[115, 116]. Han et al. placed a single IMU directly below the knee joint[117] well Shin et al placed four IMUs on both hips and feet[115]. There was no obvious improvement in performance from any of the locations.

2.4 Machine Learning

Machine Learning (ML) is a subset of computer science that focuses on systems that learn from data. An ML system can numerically estimate a complex function from exposure

to examples of a phenomenon. That is to say, an ML algorithm can convert experience into expertise or make predictions from data. As an entirely numerical approach, it is especially compelling for problems of high dimensionality. There are many different tasks where machine learning can be powerfully used, such as classification, translation, denoising and synthesis.[118, 119, 120, 121]

What separates machine learning from optimisation is that we want to minimise the task error not only for the seen experience but also for novel unseen inputs. Minimising the task error forms the crux of machine learning.[120]

The machine learning process usually follows three steps; 1) gathering a representative data set; 2) building a model to solve the task based on knowledge captured in the data set; 3) testing and evaluating model performance[121]. The output is a model that encapsulates the learnt knowledge. The model is deployable in the real world to perform the taught task.[119].

By gathering knowledge from experience, ML avoids the need to formally specify all knowledge to accomplish a task[120]. This approach significantly reduces the implementation burden and may enable previously intractable solutions for highly complex systems. Experience is provided as a set or sets of examples of the input and often the corresponding output or label. The set of examples is called the training data.

The model input for each example is a feature vector. Each element of a feature vector contains one quantitative measure of the example. Each feature is either hand picked, such as the mean of a signal, or learnt where raw data is fed directly into the model learnt. The choice of data representation or features can heavily affect the performance of a machine learning system. Hand picking features are labour intensive and without care can result in bias in the machine learning model. However, learnt features require more data for training and are harder to control.[122]

The quality of the training data is essential to good learning, as by the adage garbage in, garbage out. However, data quality measurements can be challenging because they must consider qualitative factors such as realism.

The training data is used at various stages throughout the learning process to provide knowledge and verify system performance. The whole training data is split into three sets. Training – a set of examples from which the system learns. Validation – a set used to evaluate the generalisation performance during training. Test – used to evaluate the final generalisation performance after training.

Most ML algorithms involve optimisation of some form. Optimisation refers to the task of

either minimising or maximising some function. The function we want to optimise for is called the criterion. The criterion measures what a good prediction is. When minimising the criterion, the criterion is often called the cost or loss function.[120] The learning algorithm uses the criterion to optimise model weights and biases to incorporate knowledge from the training data.

There are four standard techniques for Machine Learning: supervised, unsupervised, semi-supervised, and reinforcement learning. Each is useful for different tasks and require different forms of experience.

Supervised learning uses a labelled dataset to produce a model that can deduce the correct output from a given input[121]. In unsupervised learning, the training data set is unlabelled. The system is left to discover variation and beneficial properties across the data set. Semi-supervised learning lies between supervised and unsupervised. Some but not all example inputs have labels. The algorithm uses known inputs to label unknown examples to build a more extensive labelled training set[123] Reinforcement learning does not experience a fixed dataset. Instead, they interact with an environment using feedback to learn.

An additional form of learning is transfer learning. The research field of transfer learning is concerned with the reapplication of captured knowledge. The application changes can be significant, visually identifying a new object, or minor, such as personalisation to an individual. Many schemes exist for achieving knowledge transfer. Schemes include fine-tuning part or all of a model, extending an existing model with additional layers or generating a mapping to adapt a new input source.[124, 125]

Understanding the final performance of the trained model requires additional metrics as the loss function is often not directly interpretable. These quantitative metrics represent the ML system's ability to perform the desired task. Often the metric will be directly inheritable, such as the proportion of examples where the output was correct. The performance metric is generally evaluated for all three data sets to evaluate different aspects of model performance. For example, generalisation, or the performance for unseen data, can be evaluated using the test set.

Many issues may occur while developing an ML system. For example, over/under-fitting. Fitting problems occur when the model either learns too tightly to the training set or cannot learn the desired task. Therefore it cannot predict new unseen inputs. Adjustments to the learning rate, training data or training time will affect fitting. Adjustments to system hyper-parameters control properties such as these. Any settings determined outside the learning algorithm are called hyper-parameters.[120]

Many ML models exist and have been used widely across many different tasks. Some typical machine learning models are Random Forest, Support Vector Machine (SVM), Multi-Layer Perceptron (MLP) (Dense when fully connected), Convolutional Neural Network (CNN), Recursive Neural Network (RNN).

MLP, CNN and RNN are all forms of Artificial Neural Network (ANN). They are referred to as networks because they are typically implemented by composing together many different functions. The neural aspect comes from the original inspired by replicating biological neurons.

Multiple layers of neuron units form an ANN. The layer location in the neural network determines its name. The first network layer that receives input data is called the input layer. The network's last layer is called the output layer for similar reasons. The intermediate layers are hidden layers. By adding more layers and more units within each layer, a network can represent functions of increasing complexity[120].

Each layer in a neural network is composed of many cells or units. The cell's make-up depends on the type of layer. The connections between cells are also dependent on the layer type. All cells have at least one feed-forward connection to the next layer. A fully connected layer is where each cell connects to every cell on the next layer. When forward feed networks include feedback connections, they are called RNNs.

Two common ML architectures are CNNs and RNNs. The CNN is a specialised neural network that use convolution to combine inputs. They are well suited to grid-like data such as images or regular sampled time-series data. This architecture has been successful in practical applications, primarily visual problems.[120]

2.5 Long Short Term Memory (LSTM)

The RNN is a family of neural networks widely used in the field of translation, text, and time series prediction. Their structure allows them to process longer sequences than practical for other networks.[120] They are superior to a standard RNN system as they overcome well-documented issues such as the "vanishing gradients" problem. Vanishing gradients arise from the large chain of multiplication that occurs when performance error back propagation in an RNN.[126, 127]

The LSTM resolves some of the issues with vanishing gradients by introducing gates to control information flow[128]. During training the gates learn to forget thus to regulating information held[70]. Since conception LSTMs have become popular with many researchers as an effective and scalable model for learning problems related to sequential data.[129, 78]

There are still issues that must be considered, such as a tendency for the output to converge based on a straight pattern since the input order is chronological.[127]

The original LSTM cell was created by Hochreiter and Schmidhuber in 1997 [128] but has been improved upon by multiple researchers. The most common form used today features the forget gate introduced by Gers et al.[130, 78]. The Gers’s cell architecture is shown in Figure 2-14.

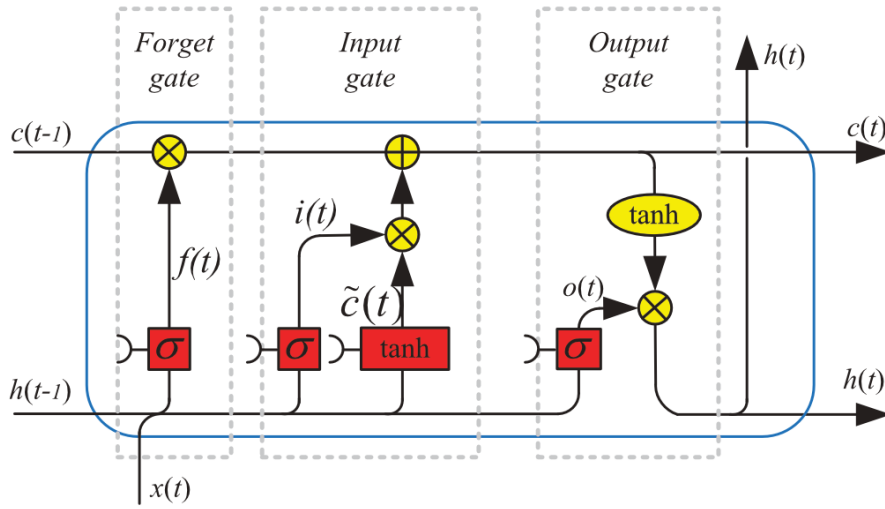


Figure 2-14: LSTM with Forget Gate[78]

An LSTM cell learns to add or removing information to the cell state, $c(t)$. This cell state is regulated using the forget $f(t)$ and input $i(t)$ gates. The forget gate determines what information to forget, while the input gate determines whether new information should be added to the cell state. So the new cell state is a function of the previous cell state ($c(t-1)$), modulated by the current input ($x(t)$) and the previous hidden state ($h(t-1)$), combined with some of the current input ($x(t)$). The output from the LSTM, termed the hidden state $h(t)$, is a function of the new cell state modulated by the output gate ($o(t)$).[127]

2.6 Locomotion Mode Recognition

Classifying human activities from sensor data is challenging. The signal difference between activities is often subtle and highly individualised.[131] Many different methods have been tested to address this problem including heuristic, statistical and machine learning methods.

2.6.1 Non-ML Methods

There are numerous forms of activity classifiers that do not use machine learning techniques. These include manual rule based heuristic methods and pattern recognition.

These have the advantage of usually being simpler both in operation and also understanding than machine learning methods. However, they normally require manual tuning and setup to adapt to an individual and struggle with higher dimensional data[12]. As well, the classification output can only be made after the rule has been completed so there is a delay in outputting a classification.

An activity recognition heuristic is usually a fixed set of rules that controls the transition between activity modes. Heuristic methods are the current standard for modes selection in powered prostheses[132, 133, 134, 135]. However there are very limited contemporary references of these methods being used.

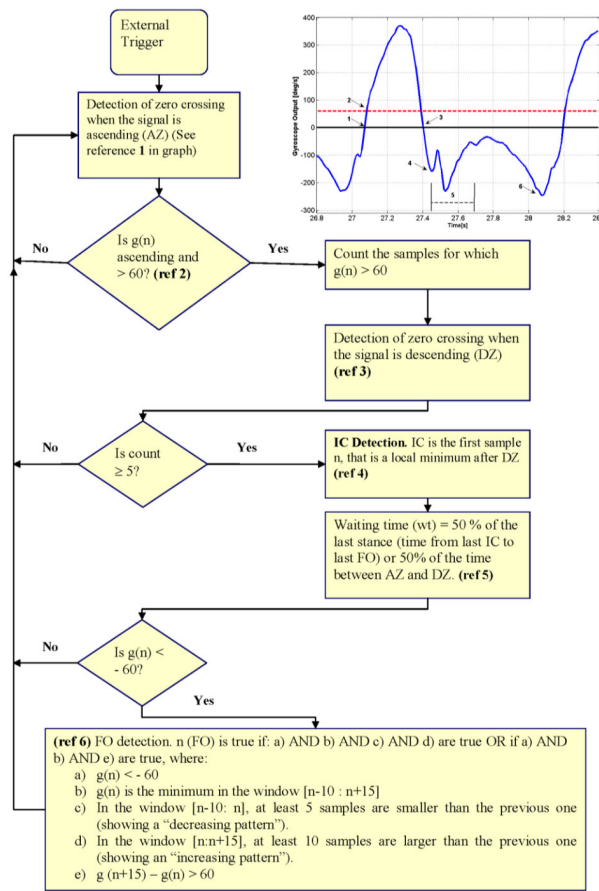
Formento et al. presented a heuristic for classifying walking, stair ascent and descent based on the output from a gyroscope. It was established by Coley et al [136] that the sagittal plane output of a shank mounted gyroscope varies during stance depending on the activity. Detection success was over 93% however the classification can only be made after the step has occurred. Figure 2-15 shows an illustration of the heuristic.[96]

2.6.2 Machine Learning Methods

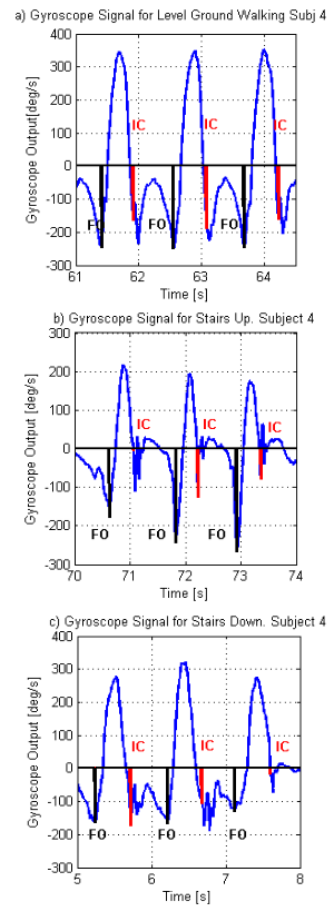
Researchers have produced many machine learning architectures for solving HAR problems[75]. Typically, HAR ML models all follow the same structure. A window of sensor data is collected and fed directly into the model. This direct approach improves the signal-to-noise ratio[70]. The model classifies the activity producing the corresponding output. Figure 2-16 illustrates this structure.[75]

Machine learning methods offer the potential to achieve higher performance, for more classes, with less manual input[70, 131, 24]. Deep learning can also provide increased flexibility, robustness and improved performance. However their learning can be difficult to control, and the solution difficult to understand[137, 129]. ML systems also require a large amount of data to train[138]. For target users whose behaviour differs significantly from the training dataset the user may suffer degraded performance. It can also be extremely laborious or impractical to collect and label large set of data, especially for the elderly and disabled people.[139]

Most researchers developing machine learning processes for HAR use a supervised learning approach[140, 141], with only a few examples of unsupervised and semi-supervised learn-



(a) Heuristic rules [96]



(b) Example input data [96]

Figure 2-15: Stair Ascent and Descent Heuristic

ing[142].

Locomotive data is usually continuous time series in nature. The continuous sensor data must be split and labelled with the current activity for supervised learning methods. The most common form of this is to use a sliding window that divides continuous data into chunks. The activities encompassed by the window determine its label.

A large number of papers use Convolved Neural Network (CNN) and Long Short Term Memory (LSTM) networks to implement HAR. Both network types are well suited to regularly sampled sensor data.

CNN architectures use convolutional and pooling layers to extract features from the sensor data. A dense MLP layer forms the final classification based on the outputs from the final CNN layer.[143, 144, 145]

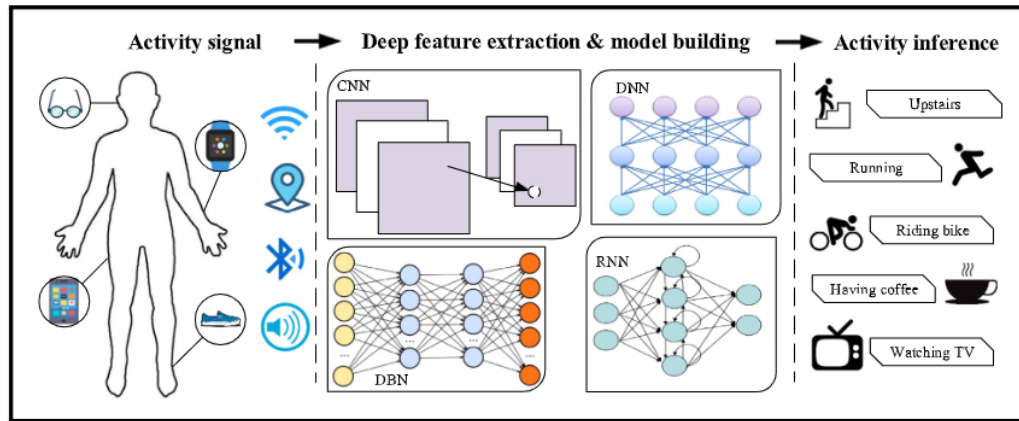


Figure 2-16: An illustration of sensor-based activity recognition using a machine learning approach[75].

RNN networks have been used frequently for HAR problems. The most commonly implementation uses LSTM cells. The layer structure of LSTM networks is such that the output of each LSTM unit layer is fed only into the LSTM units at the same time step. The formation of final classification uses either all the outputs of the last layer or only the last set of units.[146, 147]

Recently researchers have begun combining CNN and LSTM networks to form Deep Convolutional and LSTM networks[91]. The convolutional layers are placed either as the input or just before the output.[148, 149]

Both LSTM and CNN networks are used extensively throughout literature, Performance for these networks is high with most achieving $\geq 95\%$ accuracy on publicly available data sets collected in lab conditions. Data from IMU sensors is the most common input into these networks.[150, 25, 151, 149, 152, 147]

2.6.3 Personalisation Techniques

ML classifiers are constructed assuming that the probabilistic distribution between the source and target domain are equal[124]. In reality, this is never the case, so methods to account for differences between domains have been developed. Where the model is adapted to an individual user, this is typically termed personalisation.

Personalising ML models is a common issue and has been addressed in many ways across different research areas[153, 154]. Schneider et al. divide personalisation methods into two groups, shaping and data grouping[155]. In shaping the behaviour of a network is biased or shaped towards an individual, in data grouping, the target data set is enlarged by adding

data from similar individuals to it. Both of these techniques take advantage of data from others to reduce labelled data requirements for the target subject[156]. The following survey of literature will be divided into these two categories.

Shaping the behaviour of a network can occur at different times during training. Two are common, the beginning, early, or the end, late. In early shaping, the model is first trained with target data, followed by a more extensive source data set. The opposite is done in late shaping, where a general model formed from a large source training set is fine-tuned with target data. Fine tuning is performed by additional training of a pre-trained model with a different training set. This method is common and generally referred to as transfer learning.[155]

Transfer learning is the ability to extend what has been learnt in one context to another non-identical but similar context[157]. The change in context can be either the task, the domain or both. Transfer learning is appealing since it is often faster, as a model does not need to be trained from scratch for each target.

Transfer learning is generally achieved in two phases. First, a generic global model is trained from a source data set. Then it is adapted to the target by additional training using only the targets data. The influence of the target is controllable by both the number of iterations and the number of layers trained.[155, 158]

A subset of transfer learning is domain adaptation, where the domain changes but the task remain the same[120]. Domain adaptation techniques often focus on learning and applying a mapping between the source and target input data rather than fine-tuning an existing model.

Yoon et al. presented a transfer learning scheme for personalising an LSTM based language model trained to generate stylised sentence completion. Their work focuses on techniques that allow transfer learning using only a small quantity of target data and limited computing resources. Two schemes are investigated to achieve this reduction: training a new layer between the output and last LSTM layer, and freezing the first n-layer and fine-tuning just the subsequent layers. Both methods reduce the training requirements compared to fine-tuning the entire base model while achieving similar performance.[159]

Fu et al. developed a domain adaptation method for unlabelled target data denoted Joint Probability Domain Adaptation with Improved Pseudo Labels (IPL-JPDA). The method produces a transformation matrix to adapt the input data of the target to the source domain removing the need to adjust the model itself. The study collects labelled data for a set of ten subjects in a controlled environment. This data is then split into target and source

data sets with the performance tested using a cross-validation method. Personalisation is undertaken using the IPL-JPDA method and tested against a subset of the data windows for each activity. Their method achieves 93.2% accuracy, an increase of around 2% over the baseline.[107]

The other category of personalisation is data grouping. In data grouping, the target data set is enlarged by supplementing it with data from existing source data. Each individual will differ from others, but it should be expected that the population as a whole or a subset of it are similar[155, 160]. Identifying and combining similar individuals is the central area of concentration for this field.

Ferrari et al. investigated data grouping personalisation methods that weight the influence of training data based on similarity to the target subject. Individual’s similarity was evaluated by comparing physical traits (age, weight and height) and input feature vectors. Three public Activities of Daily Living (ADL) data sets were used to test performance, niMiB-SHAR[161], Mobiact[162] and Motion Sense[163]. All data sets were collected in controlled conditions. Using the weighted training data, an Adaboost classifier was trained for each target subject. The experiment was repeated with and without target data included in the training set. Excluding the target saw only a small performance improvement, compared to without similarity biasing. Including the target in the training data increased classification accuracy by $> 10\%$, on average achieving 87.39%. These results suggest that weighting the training data set towards the target subject has a larger influence on performance than similarity on its own.[164]

Nguyen et al. presented another data grouping technique using a DeepConvLSTM architecture. The model used learnt features, so determining the similarity of the feature vector was not possible. Instead, the output of the last LSTM layer was used as a pseudo for the feature vector. A Frechet Inception Distance (FID) algorithm was used to score the similarity of subjects. The score was then used to group source subjects by two schemes: selecting the closest n neighbours and clustering subjects into communities. It was noted that this correlated closely with physical characteristics. The groups were then used to train a new model from scratch and fine-tune a general model. Fine-tuning a global model proved more effective. This method improved the global model performance by 3.5% to 84.2%. The experiments were performed on four public data sets; OPPORTUNITY[165], Daphnet Gait[166], Wetlab[167] and Mobiact[162] data sets, all of which were collected in closed controlled environments.[160]

Several authors attempted to combine both transfer learning and data grouping techniques. These methods used data grouping techniques to produce a base model, which was subse-

quently fine-tuned using data from the target.

Wang et al. presents a source selection and transfer learning approach for a CNN-LSTM architecture for unsupervised transfer learning. First source subjects were selected based on a closeness score. This score combined a cosine similarity function and a hand-selected value based on the physical similarity between sensor locations. Using the selected source subjects an ML model was trained. Fine-tuning of the model was achieved by inserting and training an adaption layer between the last two dense layers. The investigation was performed using the OPPORTUNITY[165], PAMAP2[168] and UCI DSADS[169] data sets, which again were all collected in controlled conditions.[170]

Cruciani et al. presented work on personalising an activity recognition model built from the subset of a general population. The subset of subjects was selected by comparing the similarity of manually selected features. Those with the closest matching traits were used to generate the base model. Further training was then performed using a small amount of target data. This approach achieved a 5% improvement in performance when compared to selecting a source subset at random[171]. The experiment was performed on the ADL Extrasensory data set published by Vaizman et al.[172], this data set was collected using a smartphone in uncontrolled conditions with limited guidance given on how to collect or label the data.

2.6.4 Application to Amputees

The gait of a lower-limb amputee varies dramatically between individuals[173] often presenting asymmetrically[174]. This asymmetry results in a substantially different gait from the norm. Therefore techniques that work for non-amputees are not necessarily directly applicable to amputees or other gait impairments[175]. As ML performance degrades where a similar user is not in the training pool personalisation of a model is really important for amputees.

Bespoke models can achieve good classification performance for individuals, including amputees. However, there is still significant room for improvement of IMU-based LMR classifiers. The difficulty in collecting amputees' data limits research in this area and the practicality of deploying a developed system. Therefore any system that can leverage knowledge from other more obtainable sources would be highly advantageous. None of the three studies into amputees have publicly released their data.

Only a few papers have applied ML techniques to the problem of IMU-based LMR classifiers for amputees. Of them, only three papers have tested their methods on amputee gait data[144, 27, 175]. The lack of testing on amputees is likely due to the difficulties of

collecting gait data from amputees[176].

Bruinsma et al. tested different configurations of RNN and CNN networks. Gait data from a trans-femoral amputee was collected. The amputee wore two IMUs mounted on the thigh and shank. All collection of gait data was in a controlled environment. The gait data did not include any non-amputees subjects for comparison. Evaluation of different network architectures revealed that the Gated Recurrent Unit (GRU) and LSTM networks performed highest, achieving greater than 80% accuracy. The best performance of 93.06% occurred when using a GRU network fed with data from both IMUs.[27]

Su et al. demonstrated CNN classifiers with both amputees and non-amputees. Su produced an IMU dataset collected in a controlled environment to test the classifiers. Ten non-amputee and a single trans-tibial amputee were asked to walk up and down a single staircase, ramp, and flats. The IMU was placed on the healthy leg of the amputee. Both a general classifier tested with a subject not used for training and a bespoke classifier trained and tested with data from a single individual were tested. The bespoke classifier performed as highly, achieving an average performance of 94% for the ten non-amputees and 89% for the single trans-tibial amputee. The results of the general model showed a drop in performance to an average of 82%. It is not clear if this number includes performance using the trans-tibial amputee.[25]

Jamieson et al. performed a study comparing the performance of supervised classifiers for both amputee and non-amputee carrying out the same activities. Jamieson collected a dataset of IMU gait data from eight non-amputees and four subjects with lower limb amputation. Each subject wore a single hip worn IMU as they walked an improvised route through a natural environment in the vicinity of their homes. Both Support Vector Machine (SVM) and LSTM networks were trained to classify different locomotive activities. Classifiers were trained with varying sets of data constructed by Leave One Out Cross Validation (LOOXV). Several configurations of subjects were tested. Most notably, a network was trained using exclusively non-amputee data but tested using amputees. Classification accuracy fell from 78% for non-amputees to an average of 28% for the four amputees. Jamieson concluded that classifiers trained using exclusively persons without gait impairments would not be suitable for impaired gait.[175]

Lonini et al. considered a personal model trained from a target's data to be required; or alternatively, a global model trained from other similar patients. The study involved classifying physical activities based on sensor data from a waist-worn accelerometer. Eleven non-amputee and ten patients who use a Knee-Ankle-Foot Orthotic (KAFO) device were asked to perform five actions. Their work suggested that models trained exclusively from

subjects without gait impairments perform poorly when tested using the KAFO subjects. However, the model trained from only KAFO subjects performed slightly worse than a baseline personalised model.[173]

Gao et al. presented work investigating Reinforcement Learning (RL) schemes for controlling a lower limb prosthesis personalised to an individual. They first collected data from two non-amputees wearing a below-knee prosthetic device. This data was used to generate base knowledge that could be used as a starting point for their RL scheme. Using the human locomotive simulator OpenSim they implemented models of amputee gaits. These were then used to demonstrate an RL scheme that could adapt to an individual.[177]

There is limited literature on applying classifiers built from non-amputee data to amputees or those with other gait abnormalities. However, literature suggests that in order to achieve adequate classification accuracy gait abnormalities of an amputee must be taken into account.

2.7 Conclusions

Gait is a highly individual trait. The human body has evolved to use distinct control modes for accomplishing different locomotive actions. Lower limb amputees have many gait difficulties. For a lower limb prosthetic device to fully restore the lost functionality, it must be powered and replicate the different control modes. Part of the challenge of achieving this is determining the locomotive intent of a subject.

Machine learning has made significant inroads in identifying human activities from low-cost IMUs likely to be present on powered lower limb prostheses. However, limited research has occurred in machine learning techniques for human activity recognition of lower limb amputees. The lack of research is likely due to the difficulties of collecting gait data from amputees. Research into ways to reduce the data requirements for IMU-based locomotion mode recognition systems for lower limb prosthesis is required.

Chapter 3

Methodology

A new set of first-hand gait data is required to investigate activity recognition systems. Collection of this data requires the development of a data collection system. The data set will comprise labelled gait data collected in an unsupervised manner from real-world environments. Therefore the system must be simple to operate unaided, portable, and non-intrusive.

This chapter presents the experimental methodology used throughout subsequent chapters. Firstly, the chapter describes a sensing system for collection of a large-scale unsupervised human gait data set. The system will be based around the Suunto Movesense wireless IMU, described in Section 3.1, and a original Android application described in Section 3.2. This is followed by a description of the data collection process and summary of collection in 3.3. The chapter then presents the development of a data post-processing pipeline, Section 3.5 and Machine Learning (ML) methods and performance analysis in Section 3.6.

3.1 Recording Hardware

The IMU/MARG sensor was selected because it's combination of relatively low frequency requirements and flexibility in placement allow for a very low intrusion sensor that can be fitted with low skill. The MARG sensor is also a low-cost and commonly available Commercial Off The Shelf (COTS) device making procurement straightforward.

The Suunto Movesense wireless MARG was selected because it is a low-cost (£70) COTS system that can be flashed with custom code providing a flexible and powerful sensing platform. The Movesense contains a nine-axis MARG sensor, heart rate monitor, temperature

sensor, and BLE radio. It is physically small, weighing just ten grams, and allows numerous attachment configurations. Figure 3-1 shows the Movesense device. The device's rear contact act as the mounting point for attaching the device using to wide variety of straps and clips, including a heart rate strap, belt clip and elasticated Velcro strap. The datasheet for the device is included in Appendix C.



Figure 3-1: Movesense Wearable IMU[178]

From literature there is no clearly optimal placement for an IMU sensors, therefore multiple location were used. Five sensors were attached to each participant in the following locations: on the inside of both ankles using an elastic Velcro strap, on each hip using a clothes/belt clip and across the chest using a heart rate strap. The location of the sensors was selected to give broad coverage of body movements while providing easy, secure and non-invasive attachment to minimise discomfort and disruption to natural movement. Figure 3-2 shows a subject wearing the five sensors.

The onboard MARG sensor is factory calibrated therefore no sensor calibration is required. Basic verification test were undertaken to ensure compliance with the manufacturer stated performance. A small coin cell battery powers the sensor, providing multiple hours of continuous operation. A configurable low power mode can extend the usable lifetime significantly.

Suunto provides an Software Development Kit (SDK) that allows developers to reprogram the Movesense customising its behaviour. A custom program transmitted live sensors readings over the inbuilt BLE radio. The transmitted data included the MARG heart rate and



Figure 3-2: Movesense sensor attachment locations[30]

temperature sensors at 100Hz. An Android smartphone held by the test subject received this data through a custom data-logging application.

The software also implemented power management. By placing the sensors in an ultra-low power state when inactive, as detected by low readings of the accelerometer, the device can dramatically extend its battery life. The subject can wake the devices by touching both rear contacts. The power management system allowed for sharing of the sensors with trial subjects without worry about battery issues.

3.1.1 BLE Data Transmission

Data is transmitted from each Movesense wirelessly to a smartphone using the built-in BLE radio transceiver. Using the BLE notify mechanism, a custom Generic Attribute Profile (GATT) service pushes data packets to a connected smartphone. Data streaming starts when the connected phone sets a notify state in the GATT characteristic. Streaming stops once cleared. Clearing either occurs on BLE disconnection or programmatically. Data streaming is a high power state only entered during recording.

Two limits restrict the data rate that the sensor can transmit: maximum individual packet size and transmission rate. The maximum packet size is 155 bytes long. The practical limit of transmission rate is 15Hz, due to the need to transmit concurrently from five sensors. These limits require multiple IMU samples be transmitted per packet to achieve a real-time

100Hz output.

IMU data from the sensor arrives as a 32-bit floating-point number. Therefore each full sample of the nine-axis MARG sensor takes up 36 bytes. Uncompressed, the byte limit only allows for four samples; therefore, transmission requires data compression.

Compressing each measurement to a signed sixteen-bit fixed point integer allow for eight MARG measurements to be transmitted per packet. To achieve compression, each raw value is multiplied by a scaling factor before typecast to a sixteen-bit integer. This compression technique retains the sub-decimal accuracy while allowing for sufficient compression. Table 3-1 presents the sensor ranges, scaling factors and resultant accuracy of each sensor. As sixteen-bit integer values have a maximum range of -32,768 to 32,767, clipping will occur if the typecast value of the sensors exceeds these limits. The chosen scaling factor was a balance between accuracy and output range. The calculation of output range requirement was empirical.

Table 3-1: Compression of sensor readings, scaling factors and resultant accuracies. Force of Gravity (g), Degrees per Second (DPS), MicroTesla (μT)

Sensor	Sensor Range	Scaling Factor	Accuracy
Accelerometer	$\pm 16g$	256	$\pm 0.039g$
Gyroscope	$\pm 2000DPS$	32	$\pm 0.031DPS$
Magnetometer	$\pm 5000\mu T$	1	$\pm 1\mu T$

When compressed, eight MARG samples fit within one packet leaving eleven bytes available. These final bytes contain a timestamp based on the internal sensor clock, sensor temperature, heart rate, and R-R interval. The addition of these was in case of future use. Temperature, heart rate and R-R interval are only updated when they change value. The remaining byte contains an update flag for each field. Figure 3-3 shows the entire 155-byte transmission packet.

3.2 Android Application Design

A smartphone will be used as the data logging platform. To enable this an Android app was developed from scratch. The smartphone application must achieve the following requirements:

- Record sensor data from multiple Movesense sensors
- Control multiple Movesense device
- Provide feedback on the state of the Movesense sensors - e.g recording or error

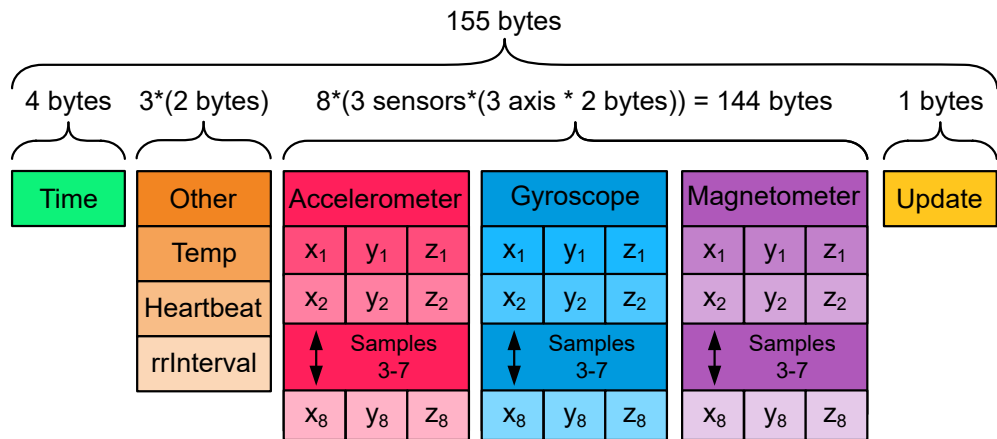


Figure 3-3: Movesense Bluetooth Low Energy transmission packet structure

- Allow live annotation of the current activity
- Share recorded data with the researchers

3.2.1 Design Process

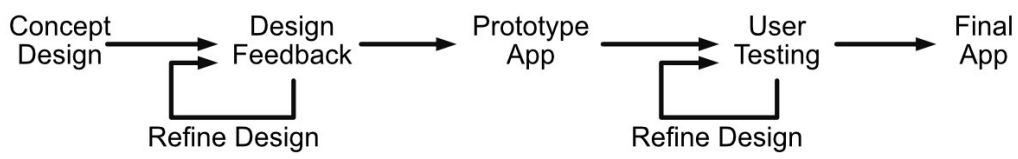
To achieve these design requirements the process show in Figure 3-4a was followed. First an initial concept was sketched, this was then refined iteratively based on feedback from other researchers. The design was then converted into a prototype android app, with a focus on the user experience over visual refinement. The user experience was refined based on feedback and observations from user testing. This feedback was then incorporated along with visual refinements into a final app used during testing. Three key stages in the apps development can be seen in Figure 3-4b.

3.2.2 Application User Experience

The application’s user interface is intentionally simple, requiring minimal instruction to use. Figure 3-5 shows the application user interface for all possible states.

When the application opens, it automatically connects to the detected sensors. This process can also be run manually by pressing the refresh button or dragging and releasing the list of devices. All detected devices show their current connection and activity states through a connection state icon and large status text. Any sensor errors are clearly shown by highlighting the sensor in red and displaying error text.

Recording begins when the test subject presses the large ‘start recording’ button at the bottom of the screen. All connected sensors begin transmitting data—their reported status



(a) Design process flow diagram



(b) Visual history

Figure 3-4: Android App Design Process

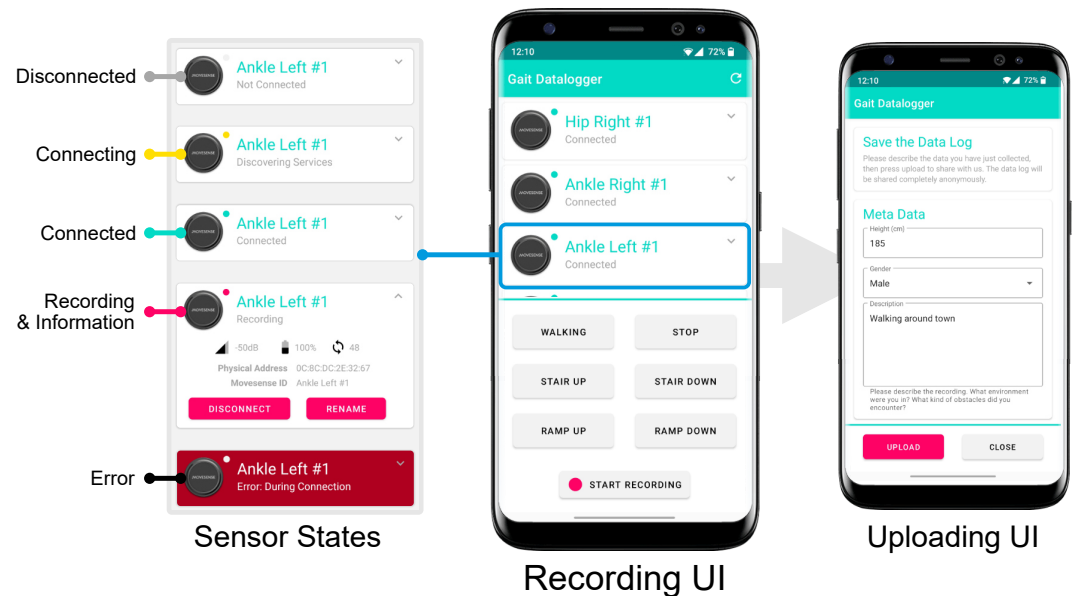


Figure 3-5: Data-logging android application user interface.

changes to recording along with a red dot icon. The onboard LED of each sensor also flashes red providing a physical indication.

When the subject presses the record button, the notify state is set on each device, starting

data streaming. The BLE service passes received data to the file saving service. This service creates plain text files locally on the phone. Sensor messages are each saved on a new line containing the sensor byte data, encoded as a hexadecimal string, the smartphone’s timestamp, and the sensors MAC address.

The subject can then begin walking around while recording data. The user presses the correct label button as soon as they change activity. Note that the application is only designed to record six activities. Activities outside of these six will be mislabelled.

When the test subject presses the ‘stop recording’ button recording stops. The app then presents the user with an upload screen. The upload screen allows metadata entry and anonymously sharing data through Google Firebase cloud services.

Figure 3-6 illustrates the interactions between each different aspect of the android application.

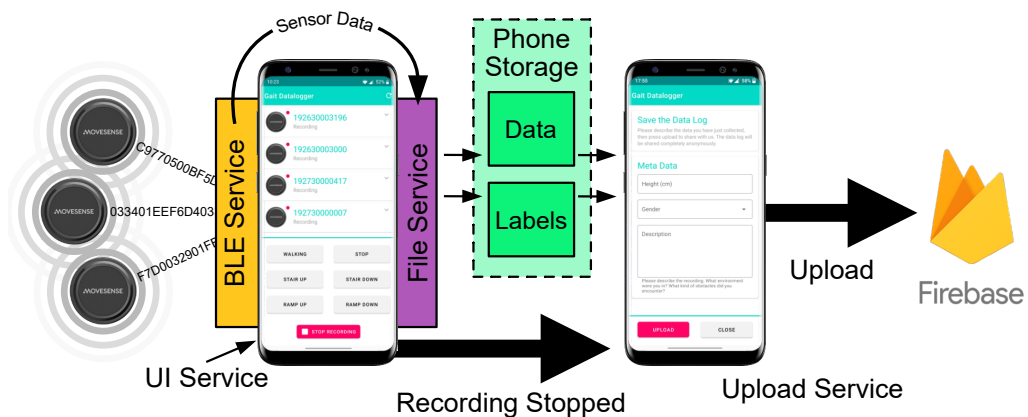


Figure 3-6: Data-logging Android App

3.3 Collection Methodology

An extensive set of gait data is required to develop ML systems for classifying locomotive mode. The gait data should be from a real-world unstructured environment which includes common imperfections and disturbances. The data will be collected using the Movesense sensors described previously. The study received ethical approval from the University of Bath Research Ethics Approval Committee for Health (REACH), reference *EP 19/20 003*. The full set of ethics forms are included in Appendix B.

3.3.1 Recording Procedure

Study subjects received instructions on using sensing equipment and general guidance on experiment procedures. The guidance comprised of general guidance to walk around a varied environment while labelling the six activity classes. Study subjects received no further instructions on recording conduct. The full test protocol is include in Appendix B.

3.3.2 Activities

The following six activities will be labelled, Walking, Stair Ascent (SA), Stair Descent (SD), Ramp Ascent (RA), Ramp Descent (RD)) and Stopped. The collection of these activities will be in the real world. Figure 3-7 shows examples of the environments for data was collected.

The collection setup only allowed labelling of the six activities described previously. Test subjects were asked to only perform the six activities but additional activities such as opening a door are inevitable. Capture of these “out of vocabulary” activities are a difficulty across all activity recognition research[179].

3.3.2.1 Effect of Ramp Inclination

There is no clear definition for a ramp in literature, but between four and twelve degrees is most common[23, 144, 110, 180]. Liu et al. found that it is possible to distinguish ramps at different inclination angles from walking. However, the steepest angles was best classified with the shallowest angle frequently confused with level walking.[110]

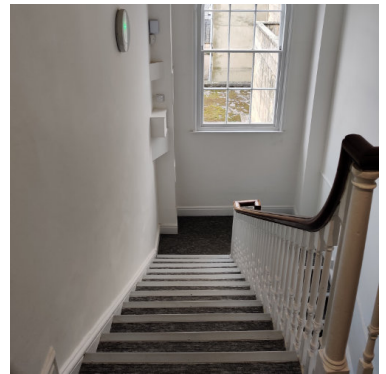
During locomotion it is not possible to measure the steepness without interfering with the recording. As such the consistency of labelling a ramp will vary between individuals and is a source of labelling inaccuracy. There is no way of tell the consistency directly from the data.

To investigate the effect different steepness of ramp have on the recorded gait additional data was recorded from known steepness slopes. Figures 3-8 and 3-9 show plots of average right ankle data during different steepness of ramp ascent. For each plot the solid line represents the mean and the filled area the standard deviation over n steps.

The plots show that ramp ascent and descent are very similar to level walking. The small differences are more prominent in the ten degree ramps than the three degree ramps. This suggests that separating steeper ramps from level walking will be easier in agreement with the result found by Liu et al[110].



(a) Walking



(b) Stairs



(c) Ramp/Hill

Figure 3-7: Example of data recording environments

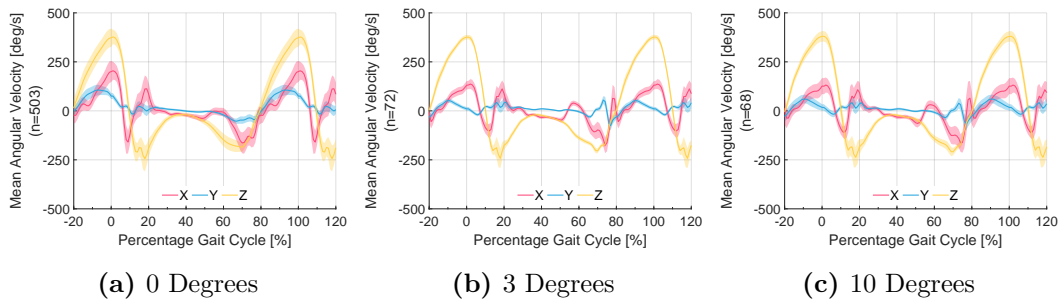


Figure 3-8: Comparison of right ankle angular velocity during ramp ascent

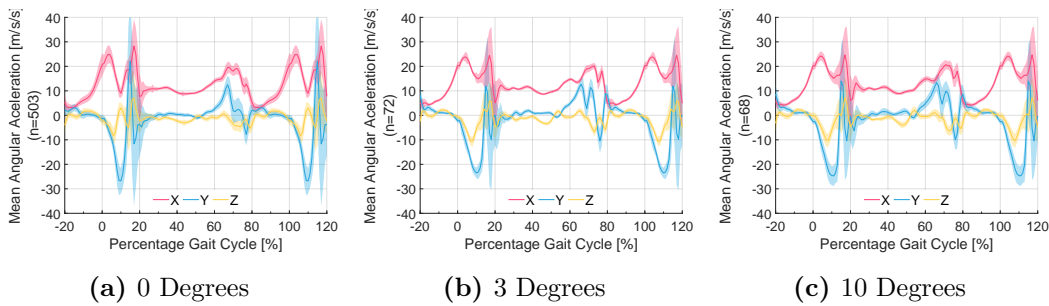


Figure 3-9: Comparison of right ankle acceleration during ramp ascent

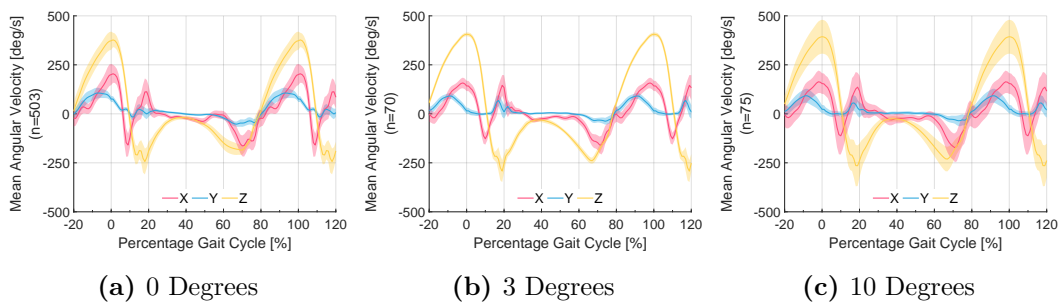


Figure 3-10: Comparison of right ankle angular velocity during ramp descent

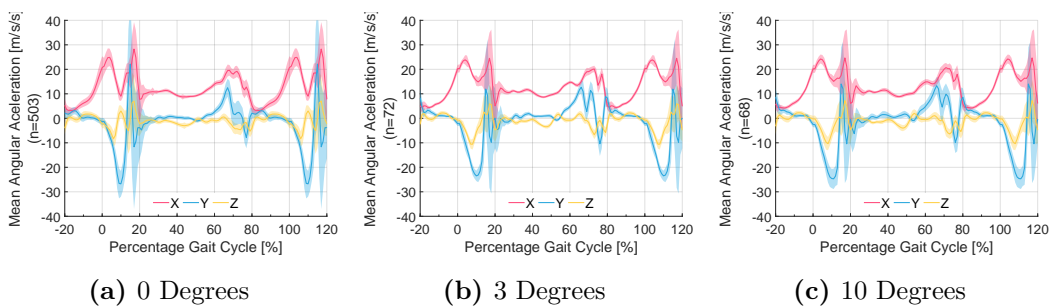


Figure 3-11: Comparison of right ankle acceleration during ramp descent

3.3.3 Data Management

Data will be managed as described within this section. This is a summary of the full data management plan included in Appendix B.

Data will be recorded as text files containing the compressed raw sensor readings from five IMUs, and a label file. This data will be shared by the test subject through Google's Firebase cloud services. Once received the data will be downloaded and catalogued. Data for each test subject and each recording will be stored in separate folders. The name of each file will contain the timestamp the data was recorded and the subject. All data will be stored as protected read-only to reduced the risk on unintended modification or deletion.

In order to minimise the chance of data loss regular backups will be taken. Data will be stored locally and backed up to a second local hard drive to protect against disk failure. A copy of the data will also be stored on the University servers. These servers are frequently backed up. Following this process will minimise any data loss and provide multiple methods for restoring lost data.

Any data collected will be published to Zenodo. All publications made will contain a data access statement with a link to the Zenodo data store.

3.4 Summary of Data

A brief summary of the data collected over the course of this research is presented below. Data was collected from a range of different individuals, a summary of the study population demographic is provided in Table 3-3. Summary of data collected is shown below in Table 3-2. Full tables are available in Appendix D. Appendix D also contains example plots of the recorded data in Section D.2.

Data was collected in three phases:

1. Large number of non-amputee participants, limited data per participant
2. Small number of non-amputee participants, extensive data per participant
3. Amputee data

The first phase of data collection focused of collecting from a broad range of individuals in different environments. Data was collected from twenty-two participants of a wide variety of age (mean 29, std 10), gender (17 male, 5 female), and physique.

The second phase of data collection involved the collection of data from a smaller number

of individuals but with a focus on collecting at least seven minutes of data for each activity. Data was collected from three subjects, two males aged 25 and 27, and one female of age 26.

The third stage of data collection focused on collecting data from amputees. The data was for one left trans-tibial individual.

Table 3-2: Summary of Data collected

	WALK	RA	RD	SA	SD	STOP
Non-Amputee total	2116564	407403	361347	278014	252763	399028
Non-Amputee Mean	96207	18518	16425	12637	11489	18138
Amputee	38114	6159	7194	2872	2450	11763

Table 3-3: Summary of Test Subject Demographics

	Age	Weight [kg]	Gender	Height [cm]
Non-Amputee Mean	28.5	16M, 6F	176	73.8
Non-Amputee Std	9.0	-	8.2	7.4
Amputee	56	Male	178	70

3.4.1 Data Specification

A summary of the the recorded data specification is provided in Table 3-4

Table 3-4: Data specification

Field	Value
Recording frequency	100Hz
Acceleration range	$\pm 16g$
Accelerometer accuracy	$\pm 0.039g$
Gyroscope range	± 2000 Degrees per Second (DPS)
Gyroscope accuracy	± 0.031 Degrees per Second (DPS)
Magnetometer range	$\pm 5000\mu T$
Magnetometer accuracy	$\pm 1\mu T$

3.5 Post-Processing

Data processing is necessary to prepare the raw recorded data for use in a machine learning environment. Within this section, the methods used to accomplish this transformation are detailed.

The data collected can be described by the hierarchical structure in Figure 3-12. Each participant has a series of gait data recordings. Each recording contains live annotation

made using the android app. Recordings likely contain different distributions of activities, each from a different environment. Each continuous period of an activity label is an episode of data. So a recording is made up of a series of contiguous episodes of data. The period around a change in the episode is a transition between activities. In the dataset, the labels represent this as a discrete change, but, in reality, it would be a smooth easing between locomotive modes.

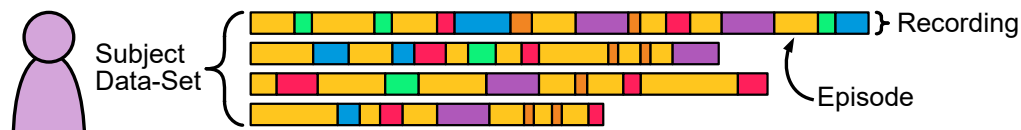


Figure 3-12: The hierarchical structure of the data recordings and terminology.

Two Extract Transform Load (ETL) scripts prepare the data for ML and address systematic issues with the data. An ETL is a common technique in data science for copying data from one or more sources to a new destination where a different representation is required. An ETL script written in Matlab 2019b transforms the sensor data from its raw form to CSV files for import into a Python environment. The second ETL script, written in Python, prepares the data for loading into the machine learning environment. The remainder of this section presents a more detailed description of both two scripts.

3.5.1 Sensor Data ETL

The sensor data ETL script transforms raw sensor data into CSV tables for import into Python. Figure 3-13 illustrates the complete ETL.

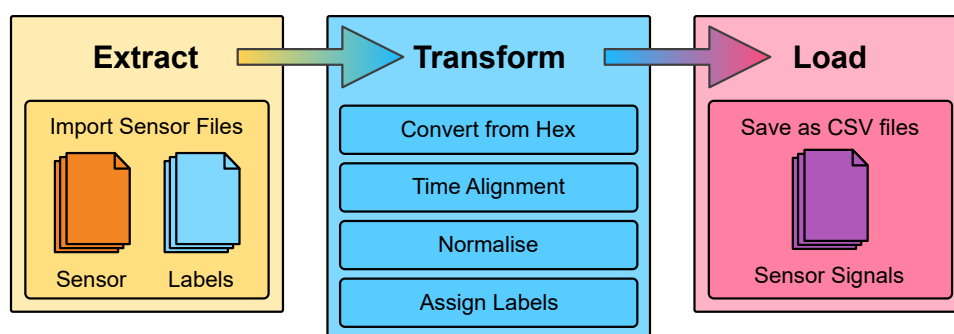


Figure 3-13: Flow Diagram of Sensor Data ETL process

3.5.1.1 Extract

Each recording produces three files; a data file, label file and meta-data file. The three files contain the following:

- **Data File** – Encoded sensor data along with the smartphone timestamp
- **Label File** – Activity labels and timestamps.
- **Meta File** – Notes about the recording, such as participant height, gender and a brief unstructured recording description. The metadata does not form part of the ETL output.

Each sub-directory is opened and processed one at a time in order of recording date.

3.5.1.2 Transform

The saved sensor data is in a hexadecimal encoding, with each pair of characters representing one byte of the sensor transmission data. The first operation is converting each pair of characters into its binary form. Then sets of binary values are typecast to integer values before applying the appropriate scalars to convert back to their original 32-bit floating-point representation. The conversion is the reverse of the on-sensor compression described previously.

Each line of sensor data contains the sensor's physical/MAC address. Files can be split into individual sensors using the MAC address. Before combining the individual sensors into a single data table, inconsistencies between the devices need to be corrected.

The sensors do not have onboard real-time clocks, with the sensor timestamp based upon the internal sensor clock. There is sufficient variation between each sensor that clock drift must be corrected. Calculations for long term drift come from comparing the sensor and smartphone timestamps. This drift is assumed to be linear; therefore, the correction offset and gain can be calculated using linear regression. Figure 3-14 shows an example of drift correction.

Each data packet contains eight sensors readings but only one timestamp. Therefore timestamps for each reading needs to be augmented. The augmented timestamps assumed a constant recording frequency.

Finally, the data is resampled to exactly 100Hz to ensure data for each sensor aligns correctly. Resampling was necessary because inconsistency in sensor clocks resulted in actual device sample rates varying by a couple of Hertz. The built-in Matlab resampling function

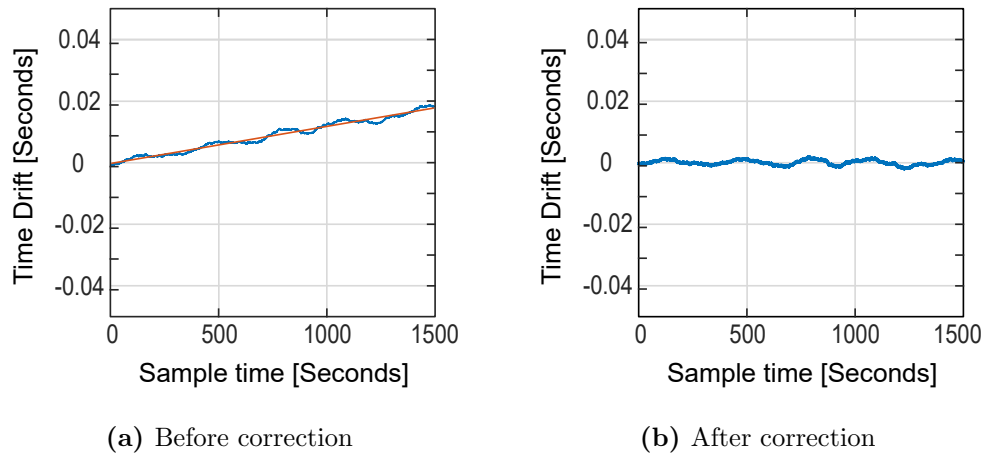


Figure 3-14: Example of sensor clock drift correction.

was used for this. The Matlab function uses a spline function to calculate the interpolated value. At this point, the data from individual sensors can be combined into a single data table.

Data normalisation is an essential pre-processing step involving scaling features to a consistent range so that greater numeric feature values cannot dominate the smaller features[181]. The main aim is to minimise the bias of those features whose numerical contribution is higher in discriminating pattern classes. Beyond the steps described above no filtering was applied to the data.

The last step involves applying activity labels to the data lines. The data labels recorded in the label file are aligned based on the smartphone’s clock. The label for each table row is set by the last activity label encountered.

3.5.1.3 Load

Two saving options were employed:

- Saving the complete recording as a single file
- Splitting the recording up into different files for each episode of an activity

Data tables are exported into CSV files, with files for each participant stored in separate folders. Basic statistics about each file include the number of samples of each activity and step count are also generated.

3.5.2 Machine Learning ETL

The second ETL script ingests the CSV data files previously generated and converts and prepares them for loading into a Tensorflow ML environment. Tensorflow requires three sets of data, train, test, and validation, each presented as a set of data inputs along with a corresponding expected output. The ETL script was implemented in Python 3.8. Figure 3-15 shows a diagram of the ETL process.

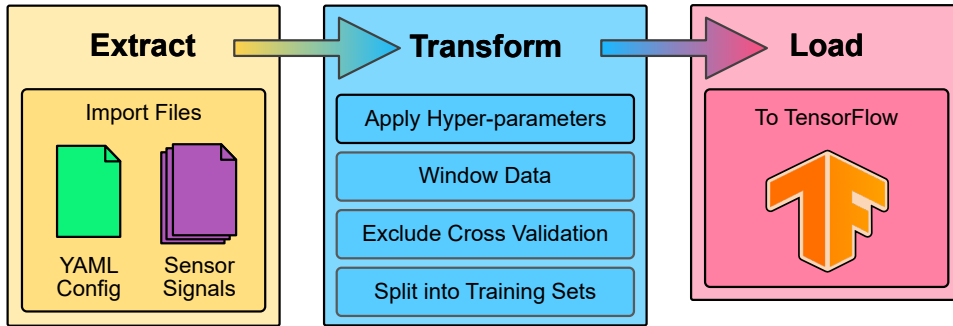


Figure 3-15: Flow Diagram of the Machine Learning ETL process.

3.5.2.1 Extract

The ETL script accepts a YAML configuration file. This file contains the configuration for the machine learning experiment. YAML files allow experiments to be replicated easily by storing the experiment set up with the input data and results. The ETL script also supports a YAML file to specify a range of values for any parameter for hyper-parameter sweeping.

The extract imports the ETL files previously generated from a directory specified in the configuration file using the Python library Pandas. The imported data is represented in Pandas data tables stored in memory, mapped to their associated participant and activity.

3.5.2.2 Transform

Hyper-parameters extracted from the YAML file are used across all aspects of the transformation process to define constants.

The YAML setup file specifies the columns of data that are required, for example, *right-ankle-gyro-y*. These data columns are extracted from the Pandas data tables, with the remaining data discarded.

Each data window contains rows of data equal to the window size. Data for each window is copied from the table to form a new table. The window selection starts at the beginning of the data table. The window starting point is moved forward by a specified skip value

for each new window resulting in a set of overlapping windows. Figure 3-16 illustrates the data windowing process.

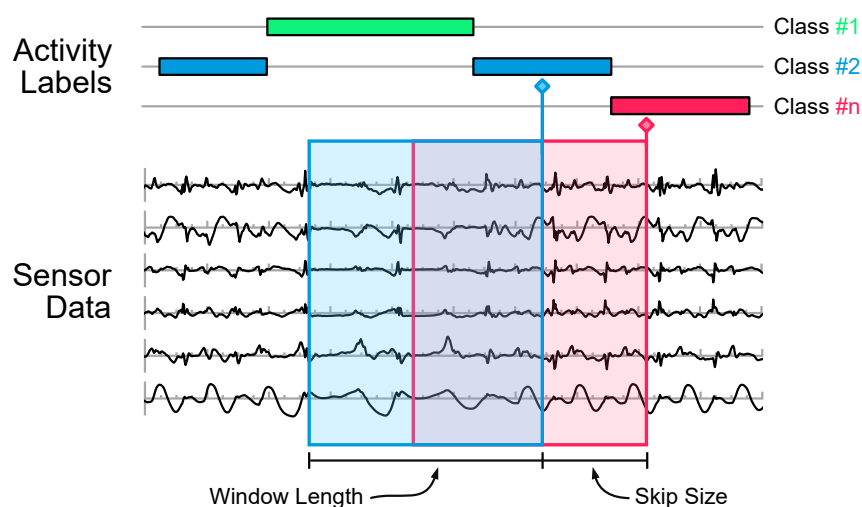


Figure 3-16: Sliding window generation

The activity labels must be provided in the same output scheme as the ML model; the output format is one-hot encoding. One-hot encoding gives each activity label an element in an array. The label array element is given a value of one; all other elements have a value of zero.

The windowed and labelled data must be split into three datasets, test, training and validation. How this is achieved will be experiment dependent; therefore, it will be discussed in the methods before each experiment.

3.5.2.3 Load

Finally, the three sets of windowed and labelled data are fed into TensorFlow. How TensorFlow is configured to process the transformed data is explained next.

3.6 Machine Learning Methods

TensorFlow[182] and the Keras[183] machine learning library will be used to develop and evaluate ML systems. TensorFlow is an open-source platform developed by Google that implements many of the workflows and tools required to develop and deploy machine learning systems. Keras is an abstraction for TensorFlow, simplifying and optimising the TensorFlow development process. This section will describe the method for generating, training and evaluating model performance.

All machine learning operations were conducted on a desktop Windows PC. The PC was fitted with an Nvidia RTX 2060 super graphics card, an AMD Ryzen 3600 CPU and 64Gb of high speed RAM.

All models will be built using the Keras sequential model framework. Keras allows for constructing models formed of a stack of layers where each layer has exactly one input tensor and one output tensor. Keras dramatically simplifies the process of implementing ML models.

The generated model can then be trained. Training will be undertaken using the Adam optimiser[184]. Two performance metrics will be used to evaluate the training performance; categorical cross-entropy for loss and categorical accuracy for classification performance. An early stopping scheme is used to end training early once training stagnation is detected in the validation data set. Stagnation is detected by a period of worse loss than the best seen.

Hyper-parameter tuning was achieved by assigning values to systematically updating model and training hyper-parameters. Some hyper-parameters configure the ML ETL, while others affect the ML model construction and training. By repeating model construction and training with different hyper-values, sensitivities could be evaluated.

3.6.1 Performance Analysis

Final model performance was conducted on the model after training. Performance is evaluated primarily by metrics derived from the classification accuracy of a test data set. Classification accuracy measures the usefulness of a model. Other performance metrics include the amount of training data required, the number of epochs to train and the size/number of model parameters. These indicate whether a model is feasible to train and deploy.

Classification accuracy is the fraction of predictions made that were correct, as shown by Equation 3-1. This is a normally presented as a percentage.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \quad (3-1)$$

The confusion matrix is another common performance analysis tool. Classification performance can be broken down by presenting the number of correct predictions for each class and where mis-classification occurred. A confusion matrix is a $n \times n$ table where n is the number of classes. The table columns represent the prediction labels, and the rows represent the actual labels. Each cell is populated with the number of classified inputs for

each combination of actual and predicted labels. The main diagonal represents the correct predictions.

From the confusion matrix additional metrics can be derived including Recall and Precision. Precision measures the proportion of actual positives identified correctly. Whereas Recall measures the proportion of actual positives identified correctly. These two values are usually plotted in a curve called a pr-curve. Often a single value is a more useful evaluation metrics than a graph. Precision and recall can be represented by the single value F1-Score.[120]

F1-score is calculated using Equation 3-2. Where t_p is the number of true positive samples, f_p is the number of false positives or other classes predicted to be the true class, and f_n is the number of false negatives or true labels mislabelled. The highest F1-score possible is 1.0 indicating perfect precision and recall.[120]

$$f_1 = \frac{t_p}{t_p + 0.5 \times (f_p + f_n)} \quad (3-2)$$

3.7 Discussion and Conclusions

Within this Chapter a new dataset of twenty-two non-amputee and one trans-tibial amputee has been collected for investigating the performance of LMR algorithms. This has been collected using a fully wireless system of IMUs and a smartphone allowing data capture to be undertaken anywhere. Data was collected in an unsupervised manner where test subjects were walk a self-selected route unaccompanied while live annotating their current locomotive activity. The data set looks promising however there are still some systematic errors that must be accounted for when using the data.

As the test subjects were unsupervised in a natural environment the experiments were uncontrolled in many aspects. This may have resulted in errors being introduced such as inconsistent alignment of sensors and labelling point. There may also be errors in the labels that will need to be identified. These errors include incorrect labels, late labelling or performing activities outside the six selected.

The data also includes data distribution issues. In a natural environment different locomotive actions are used at different frequencies. This is shown in the data with around eight times more walking data collected than stair ascent. The amount of data collected also varies per test subject.

3.7.1 Error in Performance Metrics

The performance of machine learning models is distilled down to a set of simple accuracy metrics. Metrics such as classification accuracy are solely used to compare the performance of different models. However, accuracy metrics are only as good as the quality of the test set used to generate them. There are numerous factors that can affect the quality of the test sets. The largest error is likely to be the quality of the labels. Error could be introduced into labels through erroneous input from the test subject, under-labelling where small features such as a single step are not labelled, or bias in labelling for example different interpretations of a ramp or delay in labelling.

There is also an important question over the representativeness of any test data set. A test set that consists of data of high similarity to the training set will perform better than one less similar. The selection of test data is therefore of critical importance. It is both important that it does not overlap with the training data but also that it is representative of a real world deployment of the system.

Due to the unsupervised nature of the data set calculating an absolute labelling error can not be calculated. However, the larger the test set, in both number of subjects and data per subject, the greater the mitigation against erroneous test data. Additionally careful construction of the test data sets is necessary.

3.7.2 Real Time Implementation Challenges

The ultimate aim of this research is to deploy it to a physical prosthetic device. Therefore considerations must be made as to practicality of deploying any processing step in a real time environment. Two steps require knowledge that will not be available in real time: normalisation, and drift correction. Normalisation can be replaced with either a manual fixed value or by using a form of long term averaging. Correction of sensor time drift will not be required, instead only the most recently received information will be used.

There is increasing research into deploying machine learning into low computational power embedded hardware. Classifying the current activity must be performed with low latency and in any locality therefore a cloud based solution is not suitable. Instead projects such as TensorFlow lite[185] are more applicable. TensorFlow Lite allow for pre-trained ML models to be deployed to an embedded micro-controller. This would resolve the classification issue but not training the model. It is possible that training could be performed in the cloud or on a smartphone before being deployed to the device.

Chapter 4

Understanding LSTM Locomotion Mode Recognition Behaviour

4.1 Introduction and Commentary

A large issue identified in literature was the lack of performance classification of novel unseen subjects. To address this research gap further work is required on understanding what prevents LSTM networks from correctly classifying novel subjects.

The LSTM network was selected as HAR is a time-series classification problem with, where input data that are close in space may be dependent while distant sequence of samples in time are assumed as independent. Rae et al. showed that the LSTM networks adjust well to subject-specific variations[24]. Thus, the LSTM is renowned for its performance on LMR tasks[91, 186, 187]

Following the data collection, work began on developing a general-purpose Long Short Term Memory (LSTM) network for classifying the locomotive mode of a previously unseen user. This involved work to understand how LSTM networks can recognise different locomotive modes. Once completed, an article entitled “Understanding LSTM Network Behaviour of IMU-Based Locomotion Mode Recognition for Applications in Prostheses and Wearables” was submitted to the Journal Sensors. The journal article was published on the 10th of February 2021.

The remainder of this chapter will be the presentation of the paper, followed by a short post-commentary. The paper is re-typeset to match the thesis. Page, section and reference numbering have been changed from the published copy.

Authorship and permissions

Table 4-1: Statement of Authorship

This declaration concerns the article entitled:	
Understanding LSTM Network Behaviour of IMU-Based Locomotion Mode Recognition for Applications in Prostheses and Wearables	
Publication status (tick one)	
Draft <input type="checkbox"/>	Submitted <input type="checkbox"/>
In review <input type="checkbox"/>	Accepted <input type="checkbox"/>
Published <input checked="" type="checkbox"/>	
Publication details (reference)	Sensors 2021, 21, 1264. DOI: 10.3390/s21041264 Received: 23 December 2020, Accepted: 6 February 2021, Published: 10 February 2021
Publication status (tick one)	
I hold the copyright for this material <input type="checkbox"/>	Copyright is retained by the publisher, but I have been given permission to replicate the material here <input checked="" type="checkbox"/>
Candidate's contribution to the paper (provide details, and also indicate as a percentage)	The candidate predominantly executed the: Formulation of ideas: (95%) The experiment and analysis methodology was conceived by Freddie Sherratt with supervision from Pejman Iravani Design of methodology: (95%) The experiment and analysis methodology was conceived by Freddie Sherratt with supervision from Pejman Iravani Experimental work: (95%) The experimental work was carried out entirely by Freddie Sherratt with supervision from Pejman Iravani Presentation of data in journal format: (95%) The data presentation was carried out entirely by Freddie Sherratt with supervision from Pejman Iravani and Andrew Plummer
Statement from Candidate	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature.
Signed	Freddie Sherratt Date 4th March 2021

4.2 Article: Understanding LSTM Network Behaviour of IMU-Based Locomotion Mode Recognition for Applications in Prostheses and Wearables

Freddie Sherratt, Andrew Plummer and Pejman Iravani

Department of Mechanical Engineering, University of Bath, Bath BA2 7AY, UK

Received: 23 December 2020; Accepted: 6 February 2021; Published: 10 February 2021

Abstract: Human Locomotion Mode Recognition (LMR) has the potential to be used as a control mechanism for lower-limb active prostheses. Active prostheses can assist and restore a more natural gait for amputees, but as a medical device it must minimize user risks, such as falls and trips. As such, any control system must have high accuracy and robustness, with a detailed understanding of its internal operation. Long Short Term Memory (LSTM) machine-learning networks can perform LMR with high accuracy levels. However, the internal behavior during classification is unknown, and they struggle to generalize when presented with novel users. The target problem addressed in this paper is understanding the LSTM classification behavior for LMR. A dataset of six locomotive activities (walking, stopped, stairs and ramps) from 22 non-amputee subjects is collected, capturing both steady-state and transitions between activities in natural environments. Non-amputees are used as a substitute for amputees to provide a larger dataset. The dataset is used to analyze the internal behavior of a reduced complexity LSTM network. This analysis identifies that the model primarily classifies activity type based on data around early stance. Evaluation of generalization for unseen subjects reveals low sensitivity to hyper-parameters and overfitting to individuals' gait traits. Investigating the differences between individual subjects showed that gait variations between users primarily occur in early stance, potentially explaining the poor generalization. Adjustment of hyper-parameters alone could not solve this, demonstrating the need for individual personalization of models. The main achievements of the paper are (i) the better understanding of LSTM for LMR, (ii) demonstration of its low sensitivity to learning hyper-parameters when evaluating novel user generalization, and (iii) demonstration of the need for personalization of ML models to achieve acceptable accuracy.

Keywords: Locomotion Mode Recognition; LMR; HAR; IMU; LSTM; wearables; prosthetic; prostheses

4.2.1 Introduction

For the non-amputee (in the research field this is commonly referred to as able-bodied, which can be considered an outdated term, so instead, non-amputee will be used in this article), it is taken for granted that during locomotion, both legs will act in unison adapting to the environment and activity without thought; for lower-limb amputees this ability is lost. Amputees suffer from poor gait due to muscle imbalances, and significant compensatory mechanisms are required to adapt to the loss of muscle and joints [44]. This results in musculoskeletal problems, increased energetic cost of locomotion and an increased risk of falling [6, 54, 5]. The next generation of prostheses aims to replicate the lost power generating functionality of muscles to improve gait. In order for the prosthetic to work in synergy with the user, it must recognize the users intent; therefore, a system of Locomotion Mode Recognition (LMR) is required.

Several commercially available prostheses exist that actively adapt to the user intent, such as Ottobock’s Enpower BiOM [188], Blatchford’s ElanIC [189] and Össur’s Proprio Foot [190]. None of these three provides more than basic functions, such as maintaining dorsiflexion during leg swing to increase toe clearance and adjusting ankle resistance based on terrain. Only the BiOM ankle offers powered assist in push-off, the controller for this relies on hand-tuned heuristics control strategies [191]. The University of Bath with commercial partners has also been developing a next generation powered prosthesis[78].

Machine Learning (ML) offers the ability to significantly increase the sophistication of such systems, through understanding of a wide range of activities and personalization to individual characteristics, without specialist intervention [26]. As classifying activities is temporal in nature, sequential ML networks, such as Long Short-Term Memory (LSTM), are a good fit. LSTM networks have been demonstrated to be extremely capable at Human Activity Recognition (HAR), accurately identifying actions from locomotive actions, such as Walking, Running and Stairs [150], to Hip-Hop dance moves [192]. However, little is known of their understanding internal behavior during these tasks. For a medical device, such as a prosthetic, both a high levels of accuracy and detailed knowledge of internal network operation is required.

This paper explores in detail the operation and performance of LSTM networks for LMR using both seen and novel users. Data from non-amputee participants is used as a substitute for amputee data as it allows for a much larger and more varied data set while minimizing risk to subjects. This is then used to investigate the internal operation on a simplified LSTM network. The effects of hyper-parameters on the generalization performance of a practical LSTM network are then investigated. Finally, changes to the model are investigated to try

and improve its performance for novel users.

The major contributions of this work are as follows:

1. Methodology for the collection of a large self-supervised data set of human locomotion data in a natural environment.
2. Provide an insight into the behavior of an LSTM LMR model, and the performance effects of hyper-parameter selection.
3. Investigation of hyper-parameter sensitivities in an LSTM network and their effect on classification accuracy and generalization to novel user.
4. Demonstration of the need for personalization techniques to account for individual gait traits.

The remainder of this paper is organized as follows; First background theory on the Human Gait Cycle and LSTMs is presented in Section 4.2.2. Section 4.2.3 contains Related work followed by Section 4.2.4—Materials and Methodology, describing the data collection process and setup of the ML environment. The following Sections 4.2.6 and 4.2.7, detail the experiments undertaken, investigating LSTM behavior, and hyper-parameter sensitivities, respectively. These each follow the same structure with an introduction to the experiment, analysis methodology, then results and discussions. The remaining two sections, Sections 4.2.8 and 4.2.9, contain discussion and conclusions.

4.2.2 Human Gait and Machine-Learning Fundamentals

Within this section fundamental theory of the human gait cycle, and Recurrent Neural Networks (RNN) and LSTMs is presented.

4.2.2.1 Locomotion Mode Recognition and the Human Gait Cycle

Human gait is a cyclic process that can be delineated by key events. A gait cycle is defined by two successive Initial Contact (IC) events (the point at which the foot contacts the ground) of the same limb. As this is normally the heel, it is often referred to as Heel Strike (HS). Conversely, the point when the foot leaves the ground is referred to as Toe Off (TO). These two events are used to subdivide the gait into two phases; stance—when the foot is on the ground, and swing when not. A diagram showing these events and their location in the gait cycles is shown in Figure 4-1.

It has been shown that gait events can be established from only extrema of the shank angular velocity in the sagittal plane (The sagittal plane divides the body into left and

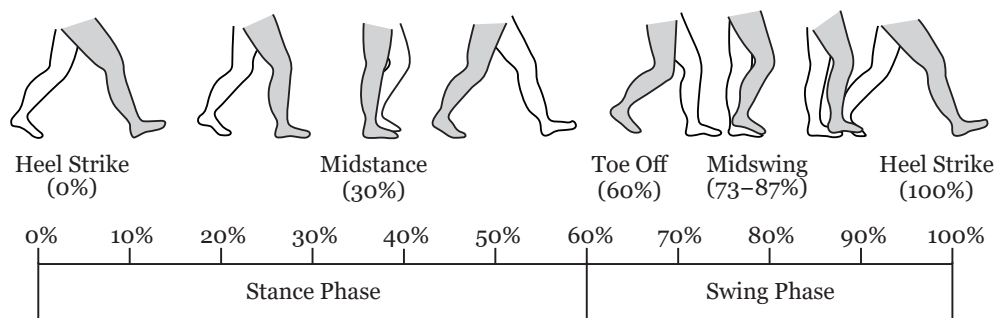


Figure 4-1: Human Gait Cycle during level walking. The percentage timings of the gait events are approximate, they vary depending on the individual and environment.

right, so rotation in this plane is forward and backward motion of the shank) using a technique originally presented by Sabatini et al. [193]. IC/HS was found to line up with the minima following the peak swing velocity (PK) and TO was identified as the halfway point between the zero-crossing, negative to positive, and the minima before peak swing. Figure 4-2 shows the gyroscope trace of a sensor attached to a subject’s shank with the locations of the calculated TO and IC events indicated.

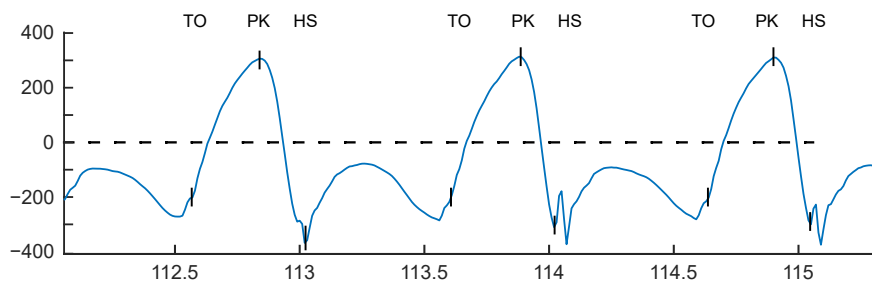


Figure 4-2: Gait events extracted from the sagittal plane gyroscope signal. IC—Initial Contact, PK—Peak Swing, TO—Toe Off.

The action of the leg varies depending on the activity. To accommodate this, powered prostheses will require multiple locomotive modes to achieve the different timing and power requirements. Therefore, automated recognition of the user’s intentions and subsequent selection of the corresponding locomotive mode will be crucial to the performance of devices [12, 194, 195]. In order for amputees to have confidence in a prosthetic device, its activity recognition must be timely, accurate and consistent and able to account for the individual gait characteristics [196, 197, 198].

For the current generation of prosthetic devices, this is achieved through hand-tuned heuris-

tics. These methods identify and associate changing properties of sensor data with different activities. For example, Coley et al. noted the variation in shank sagittal plane rotational velocity that occur when walking on stairs [136]. It was found that during early stance there is an increase in rotational velocity during stair descent and a decrease during stair ascent when compared to level walking. The current state of the art in LMR uses ML methods to accomplish activity recognition; these techniques will be discussed further in the next section.

4.2.2.2 Long Short-Term Memory Networks

LMR for active prostheses has conventionally been achieved through heuristic methods with handpicked features that are manually tuned for each individual [199, 200]. This approach is favored by the commercial market due to safety and regulatory concerns [201]. The tuning of these controllers is time-consuming and requires a highly skilled prosthetist. In the current state of the art for LMR techniques, the focus has been on the use of ML techniques to automate the process of feature selection, output classification, and personalization [26].

Many different machine-learning techniques have been investigated including, Support Vector Machines, Hidden Markov Models and Convolution Neural Networks (CNN) with success [26]. As sensor data from human gait is temporal, the best architecture for solving this will be one that can take into account the sequential nature of the input data. The Recurrent Neural Network (RNN) is an ML architecture suited to handling sequential data as it contains both vertical and horizontal connections. This means that cell activation is related to both the previous time step and the input. Information is therefore passed along the sequence as well as up through layers. Figure 4-3 shows the unfolded structure of a recurrent network. It can be seen that the activation of each cell is dependent on both its inputs and the hidden states of the previous time steps.

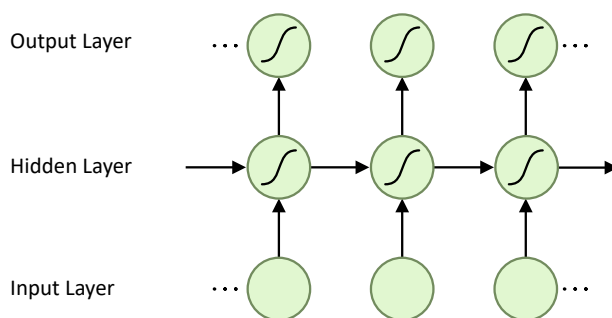


Figure 4-3: Unfolded Recurrent Network.

Each timestep in the network can contain several hidden states or units. This is represented by Equation (4-1) showing the activation input, \mathbf{a}^t . \mathbf{a} is formed from the bias vector \mathbf{b} plus the sum of input vectors \mathbf{x} and previous hidden states \mathbf{h} , multiplied by the weight matrices \mathbf{W} and \mathbf{U} for hidden-to-hidden state and input-to-hidden state connections respectively [120]. The shape of an RNN network is often described by its timesteps and units, for example, 128×6 .

$$\mathbf{a}^{(t)} = \mathbf{b} + \mathbf{W}\mathbf{h}^{(t-1)} + \mathbf{U}\mathbf{x}^{(t)} \quad (4-1)$$

RNNs have been shown to produce good results in some sequential tasks, but their application is limited by difficulty of training. The primary difficulty is the vanishing/exploding gradient problem. During gradient-based training methods, repeated multiplication by values that are not near one, along long dependency chains results in values that either vanish or explode. A vanishing gradient makes it challenging to know which direction the parameters should move to improve the cost function. Exploding gradients can make learning unstable. Non-gradient-based training has been tried, although to limited success [202, 120].

The Long Short-Term Memory (LSTM) architecture solves the vanishing gradient problem by adding mechanisms for regulating information allowing it to be retained for long periods. Created by Hochreiter and Schmidhuber in 1997 [128] the LSTM is an RNN style architecture that includes gates to control information flow between cells, see Figure 4-4. Information flowing along the cell state can be modulated by the input and forget gate structures with the final output a filtered version of the cell state based on context from the inputs [203].

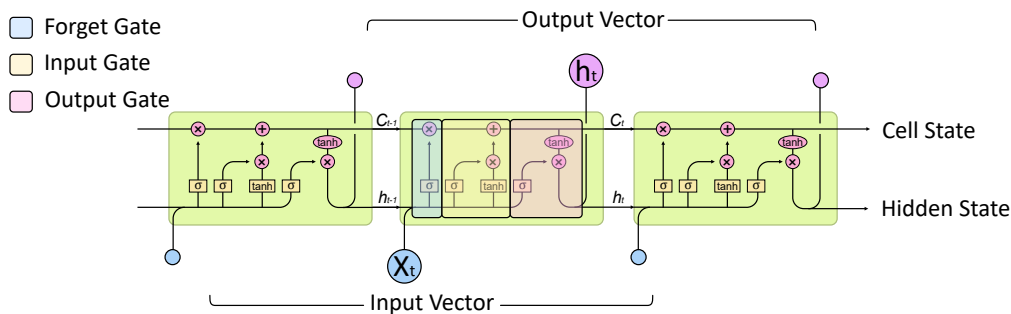


Figure 4-4: LSTM unit with input and output connections.

4.2.3 Related Works

In HAR tasks, LSTMs have been demonstrated to provide exceptional performance [150] although very little work has been done investigating this in the context of prostheses. Labarrière et al. conducted a systematic review of the ML methods used in activity recognition; for assistive device LSTM networks were only used once [26].

LSTM networks have been found to perform highly in HAR and Activities of Daily Living (ADL) tasks. Murad and Pyun investigated Deep LSTM networks for LMR [150]. They trained their network on common ADL datasets, presenting performance in comparison to other ML architectures on the same data sets. The network they used took raw IMU data as its input, then interpreted the data using four LSTM layers before a late fusion dense layer and a SoftMax classifier were used to produce a class output. The number of units in the LSTM layers was not explicitly stated but appeared to be one. Performance is high achieving 96.7% accuracy on the UCI-HAD dataset [204] and an improvement on the presented previous classification attempts using CNN, SVM and other networks. Tufek et al. replicated this result, achieving 93% accuracy on the UCI-HAD data set using only a three-layer LSTM network [151].

However, the accuracy presented is determined from the validation data, a random 20% of the source data, so sufficient separation between training and validation data is not guaranteed. In the compared work, a mixture of evaluation techniques is used, most commonly k-fold cross-validation techniques. With test data selected by leaving out participants [82, 205]. As such, it is not clear that a direct comparison can be made to demonstrate LSTM's superiority.

Different sensor fusion approaches have been tried. Murad et al allowed a deep LSTM network to learn to fuse the sensor modalities [150]. Chung et al. used an ensemble voting arrangement, where each channel modality of sensor data was passed through a separate LSTM network, with a weighted voting system forming the output classification [109]. This achieved a slightly higher accuracy, of 94%, than using the sensors individually.

Multiple authors have developed models that use a series of CNN layer first to fuse sensor data from multiple modalities before passing it to a LSTM network [206, 207, 148, 87, 208]. These achieve only minor improvements in performance classification with 95–96% accuracies. Again, none of the authors were clear about the unit shape of their LSTM networks.

There are few examples of LSTM networks being used in assistive devices. Wang et al. used a Deep LSTM network to select locomotion modes for a lower extremity exo-skeleton [209].

Five locomotion modes were classified (sitting, standing, walking and ascending/descending stairs) based on angular information from hip, knee and ankle joints. A two-layer LSTM network with 128 timestep windows was used. The hidden states of this were fed into a weighted mean before a SoftMax classifier. Again, the number of units per timestep was not specified. The classifier performed better than the other models tested achieving over 95%.

Ben Yue Su et al. presented work investigating intent prediction for trans-tibial amputees using IMU data and a CNN networks [25]. Ten non-amputee and one trans-tibial amputee were asked to perform short walks traversing a short staircase and ramp with a level surface either side. The non-amputee subjects wore a hands-free crutch to simulate amputation. Three IMUs were attached to the thigh, shank and ankle of the “healthy” leg. The CNN classifier identified five steady states and eight transitions between states. An accuracy of 94% was achieved by the non-amputee subjects; this dropped to 89% for the amputee for validation data. When testing generalization to an unseen user, using Leave One Out Cross-Validation (LOOXV), this dropped to 82% for non-amputee subjects. Subject-specific training was recommended. Reasons for poor generalization were not investigated.

Research into the generalization of ML HAR Models to new users is limited. Dehghani et al. investigate the metrics used to evaluate the performance of classifiers, particularly regarding their performance on unseen data presented using k-fold cross-validation methods [210]. The paper implements various forms of ML, such as Support Vector Machines (SVM) and Hidden Markov Models (HMM) but not LSTM. Dehghani found that using validation data to evaluate performance overestimates accuracy by 10–16% as the validation data is too similar to the training data. Instead, individual subjects should be excluded and used as test subjects. The reason for the worse generalization when presented with a novel user has not been investigated.

Investigations into LSTM networks for HAR/LMR have been primarily focused on achieving the highest possible classification accuracy. No one has investigated the internal operation of the network, or sensitivities to hyper-parameter selection for these applications. Dehghani et al. identified that model generalization to novel users is an area that also needs further investigation [210]. This paper aims to address these areas.

4.2.4 Materials and Methodology

To complete the aims of this paper, a dataset of human locomotion, methods for processing this data and a ML environment are required. This section details the methodology used to provide this. It is split into three sections, Sections 4.2.5 and 4.2.5.1 detail the data

collection and pre-processing, respectively. Section 4.2.5.2 presents the ML environment and methods.

4.2.5 Unsupervised Data Collection in Dynamic Natural Environments

There are several commonly used data sets for LMR of non-amputees. The OPPORTUNITY activity recognition dataset [211] contains 18 classes for Activities of Daily Living (ADL) such as opening/closing doors and drinking from a cup. Each subject wore seven 6-axis IMUs and 12 3-axis accelerometers while they performed the prescribed actions. The UCI-HAD dataset [204] recorded subjects performing six activities: walking, stair ascent, stair descent, sitting, standing and lying while wearing a waist-mounted smartphone with onboard Magnetic, Angular Rate and Gravity (MARG) sensors. Both of these data sets were recorded in controlled conditions, so do not capture any variation in the activity that may occur due to the environment. Sztaylor and Stuckenschmidt collected data from 15 subjects performing eight activities while wearing six wearable sensors. Recording took place in the same natural environments for each activity. Only steady-state activities were captured and not the transition between them [212]. Due to limitation in the identified data sets, a new set of data is required.

The aim of the new data set was to record natural locomotion in an unstructured environment, capturing both steady-state and the transition between activities across different settings from a wide range of subjects. Collection of large quantities of data from amputees is very challenging, so instead non-amputee subjects are used. Non-amputee subjects have a less varied gait than amputees, but this can be countered by a larger population size.

Non-invasive wearable sensors, such as Inertial Measurement Units (IMU), are an appealing choice for developing such a system. IMUs give fast update rates, 100s of Hz, are non-invasive (small with minimal mounting constraints), low cost and have reasonable accuracy. They have been widely used in the field, all of the latest generation of powered prosthetic knees investigated by Fluit et al contained IMUs [201].

The Suunto Movesense wearable IMU was used to collect activity data. This is a COTS device containing a nine-axis MARG sensor and a Bluetooth Low Energy (BLE) radio in a small 10 g package. The sensor housing contains a snap connector allowing it to be clipped on attachment hardware. A variety of mounting hardware is available off the shelf. The sensor is user-programmable allowing customized behavior through the Movesense API. To enable the desired streaming application it was programmed to transmit compressed IMU data at 100Hz over its BLE connection to a custom app running on an android smartphone. The devices come with Factory calibration for the IMU, no additional IMU calibration was

undertaken.

Five sensors were attached to each participant in the following locations: on the inside of both ankles using an elastic Velcro strap, on each hip using a clothes/belt clip and across the chest using a heart rate strap. The location of the sensors was selected to give wide coverage of body movements while providing easy, secure and non-invasive attachment to minimize discomfort and disruption to natural movement. Figure 4-5 shows a subject wearing the five sensors.



Figure 4-5: Subject wearing the Movesense IMU sensors on both ankles, hips and the chest.

To record data from the sensors, a custom android app was created. This formed a BLE connection to each device and saved the streamed data. During recording a series of buttons at the bottom of the screen could be used for real-time labelling of activities. Once recording had finished the subject was presented with an upload screen allowing metadata to be added. The file could then be shared anonymously with the researchers using Google’s Firebase cloud services. A screenshot of the app in recording mode is shown in Figure 4-6.

Study subjects were provided with instructions on how to use the sensing equipment, and the activity classes, then allowed to record as they wished. The following activities were selected, Walking (W), Stair Ascent (SA), Stair Descent (SD), Ramp Ascent (RA), Ramp Descent (RD) and Stopped (S). Labarrière et al. identified these as the most commonly investigated and they require no equipment or skill to perform [26]. The study received ethical approval from the University of Bath Research Ethics Approval Committee for

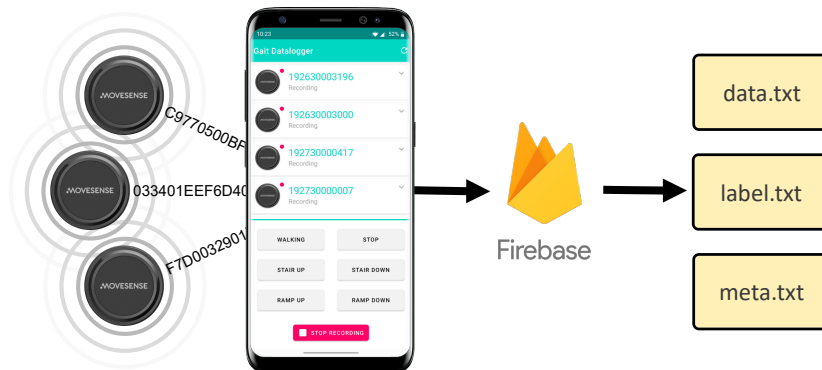


Figure 4-6: Custom Android app with connected sensors and illustration of Firebase upload system.

Health (REACH), reference *EP 19/20 003*.

Twenty-two participants of a wide variety of age (mean 29, std 10), gender (17M, 5F), and physique were chosen to give a broad data set. Participants were instructed to walk around a varied environment with the sensor on while labelling the six activity classes. No further instructions on how the recording should be conducted were provided. A total of 268 min of data was collected, which includes 1170 transitions between activities. Table 4-2 contains a summary of the data collected. The number of steps was produced by summing the peak swing gait events for each label.

Table 4-2: Quantity of data collected for each activity.

Activity	Samples	Time (min)	Number of Steps
Walking	1075211	179	9438
Stair Ascent	139922	23	1286
Stair Descent	122379	20	1280
Ramp Ascent	73328	12	656
Ramp Descent	79436	13	754
Stop	121027	20	-
Total	1611303	268	13414

4.2.5.1 Data Pre-Processing

To convert the raw saved data to a form that Tensorflow could import, a processing pipeline was developed in Matlab 2019b. The pipeline consisted of a decoding, re-sampling, time alignment, normalization, and exporting steps. A flow diagram of this process is shown in

Figure 4-7, further details of the process are described below.

The smartphone app does not interpret the compressed data stream, only saving it to a log file. Therefore, the data files need to be converted from a compressed fixed-point form back to their original floating-point representations. This is done by applying the reverse scaling factor to that used by the Movesense device to compress the data. The scaling factor was chosen to provide a balance between accuracy and compression.

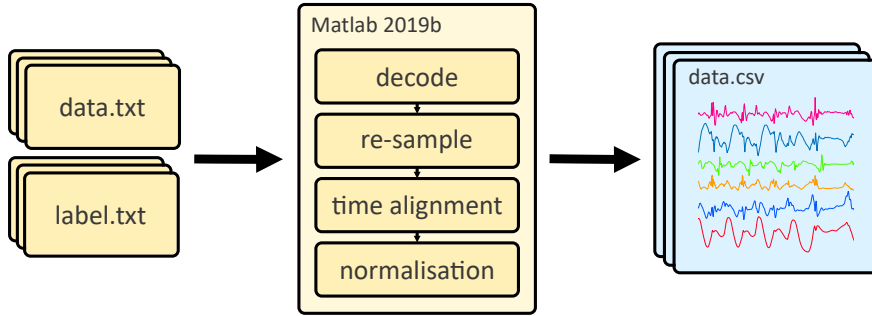


Figure 4-7: Raw data input and pre-processing flow diagram.

To compensate for the difference between the internal sensor clocks the data is re-sampled using the smartphone clock as a common reference. Once a consistent frequency for all the sensor data is achieved, this common reference allows for data from all sensors to be aligned accurately.

Finally, the data is normalized using Equation (4-2) to scale and shift the data. After this each data channel has a center of zero and standard deviation of one. Normalization is applied on an individual data file basis. In Equation (4-2) μ is the sample channel mean and σ the sample channel standard deviation. x is the input sample and z the normalized value. The normalization process removes any overall bias in the IMU data. No additional filtering was applied to the raw data before it was fed to the machine-learning models.

$$z = \frac{x - \mu}{\sigma} \quad (4-2)$$

The following axis system will be used when presenting and analyzing the results. The axes use a right-hand system with direction, front left and up for x , y and z respectively. x is forward towards the front of the body, y toward the left and z upwards. From this point on, the beginning of the gait cycle, 0%, will be defined as the peak swing maxima. This leads HS by about 20%.

4.2.5.2 Machine-Learning Methods

Within this subsection, the methodology for setup and training of the machine-learning models are presented. TensorFlow 2.1 was used, with the Keras API used to setup, train and evaluate the different ML Models. The ML environment was developed and run in an Anaconda Python 3 environment. A conventional supervised training setup was used.

Model Setup Two different model architectures were developed, a simplified model with a single information path for investigating LSTM internal behavior; and a full complexity practical model based on the architecture presented by Murad et al. [150] for investigating hyper-parameter sensitivities. This design was previously discussed in Section 4.2.2.2.

For both architectures input data is fed directly into the first LSTM layer. For models with additional LSTM layer, the full output of the first LSTM layer is fed into input of the next layer and so on. The output from the final LSTM layer is fed into a fully connected dense layer followed by a ReLU classifier. For the simplified model the LSTM output is the last timestep only, for the full complexity model the full output of all timesteps is used. The size of the dense layer is equal to the number of outputs from the last LSTM layer. A one-hot classification output is used to encode the activity classes. Figure 4-8a,b show the architectures of the simplified and full complexity model, respectively.

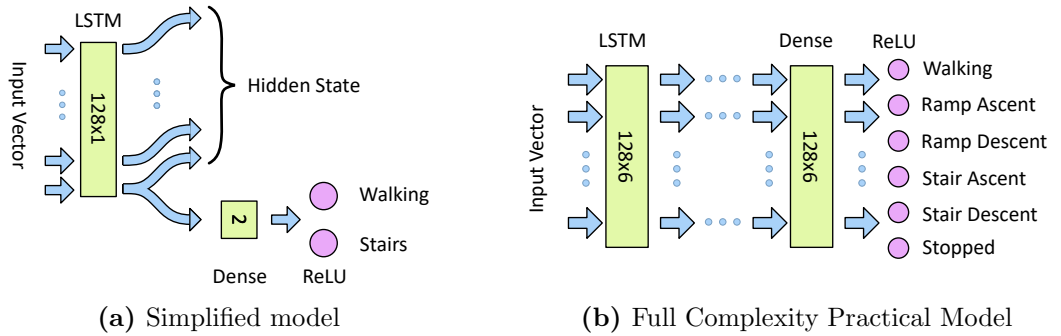


Figure 4-8: Machine-Learning Model Architectures.

Data Segmentation The data set was divided into two groups for test and training. The training set was used during the learning process with the test set reserved for evaluating the performance of unseen data. The test set was a variation of Leave One Out Cross-Validation (LOOXV). LOOXV involves training and analyzing the model multiple times with different excluded individuals, the results are then combined to improve statistical certainty. For this paper four/five subjects were excluded each time with analysis repeated five time, meaning each subject was excluded once. The training set contains the remaining

subjects, with 30% of the data used as a validation set. Figure 4-9 provides an illustration of how the data is divided between the three data sets.

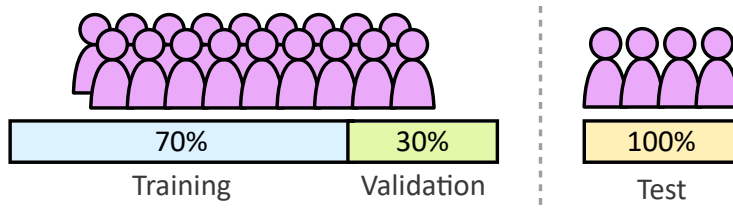


Figure 4-9: Division of the subject population to form the training, validation and test sets

To balance the data set, both the training and test sets were adjusted by removing data so that no class contained more than 50% more samples than any another. This re-balancing was undertaken carefully so that during validation splitting the balance was maintained. A class weight input was used to bias the training to further balance the class labels.

The continuous sensor data was segmented using sliding windows. Between the start of each window, an offset of five samples was used. This offset was set empirically to give the model a wide range of data windows position without slowing down learning from an unnecessarily large training set. The output label for each window was set as the recorded ground truth at the end of the window. Classification labels were presented using one-hot encoding.

Model Training The models were trained to minimize categorical cross-entropy. Model weights were initialized with a Golorot Uniform initializer [213] and optimized with an ADAM optimizer [184]. A dropout of 0.5 was used, selected experimentally, with network connections dropped between the last LSTM output and the dense classifier.

During trained the full training dataset was used for each epoch, with data passed to the optimizer in mini batches of 2000 windows. At the end of each epoch the entire validation set was evaluated. Early stopping was used to prevent over-fitting, this stopped training when stagnation of validation cross-entropy loss was observed. Stagnation was identified by three consecutive losses of greater than the minimum previously seen.

The model was trained on a PC with an AMD Ryzen 3600 CPU and a Nvidia Geforce RTX 2060 Super. Using GPU training, each epoch took approximately 10 s with between 30 and 100 epochs required to train each model depending on the model size and the number of output classes.

4.2.6 Investigation of LSTM Behavior

An understanding of the internal operation of an LSTM LMR network is important in assessing the network limitations. To capture the internal behavior the effects of input data on the output must be established. This will be achieved by mapping changes in internal hidden state to incoming data.

This analysis can only be performed on low-complexity networks, as information paths of large networks become too convoluted making tracing infeasible. The experiments will use the simplified model, described in Section 4.2.5.2, as this only has one path for information to flow along. For the simplified network analysis only a single shank IMU sensor will be used. From visual inspection this showed the most variation between activities and subjects.

4.2.6.1 Analysis Methodology

To enable changes in the hidden state to be mapped to features of the input data, typical plots of input sensor data for different activities are required. This will also allow differences between individual's gait to be assessed. A typical gait cycle was produced by combining multiple gait cycles for different activities. Each gait cycle can then be normalized to percentage gait, with the mean and standard deviation of multiple cycles plotted to produce activity trends.

Using an extraction of the hidden state, a measure of information gained from the input data will be drawn. Due to reduced learning capacity of the model, in order to get a meaningful classification accuracy, the classification is performed on a reduced number of classes. The data labels are reduced to include only the three most prevalent activities (Walking, Stair Ascent and Stair Descent). The total output classes can be reduced further by combining pairs of these.

Four different combinations of output class for the three activities were tested with four different combinations of input sensor, y Gyroscope, x Accelerometer, y Gyroscope and x Accelerometer, and a full six-axis IMU. The y gyroscope and x accelerometer were selected as visually they showed the greatest variation between activities.

To extract the hidden state the weights and biases of the trained LSTM layer were extracted and copied into a new model whose output was the full hidden state sequence. Input data was then fed into the new network to extract the hidden state. To observe patterns in the hidden state, multiple data windows were overlaid. Variations in step cadence were removed by normalizing to gait cycle. Different activities were then plotted independently to show how the network acts to each. The hidden state output is shown on the y axis. This is

a dimensionless value which tends toward a value depending on the network classification decision. The value is dependent on the classifier weights but is typically -1 , 0 or 1 . The final element of the hidden state is fed into the dense layer which forms a classification based on its learnt thresholds.

4.2.6.2 Results and Analysis

Below the results of the experiments investigating the internal behavior of a LMR LSTM network are presented and analyzed.

Individual's Gait Trends Figure 4-10 show the typical sensor data trends for different individuals and activities. The solid lines represent the mean and the filled area the standard deviation. the x accelerometer and y gyroscope signals, for three different activities W, SA, SD. On each plot three individuals have been super-imposed over each other. 0% gait cycle corresponds with the peak y -axis shank angular velocity.

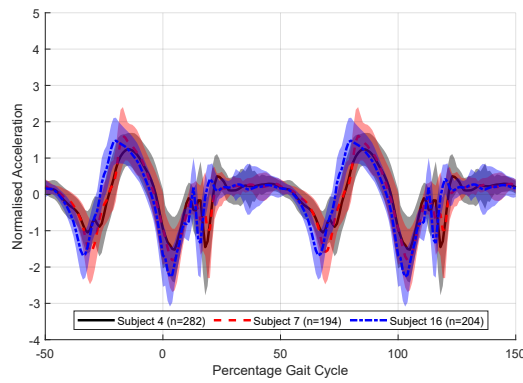
From Figure 4-10 the differences between the three chosen participants can be seen. The x acceleration signal is very noisy, with large standard deviation seen particularly around heel strike, 20% gait cycle. Smoothing of the input data could be used to reduce this, but this was not investigated. The gyroscope signals are more consistent, shown by the reduced standard deviation.

The stance angular velocities match the results presented by Coley et al. [136]. Stair ascent has a lower early stance rate and stair descent a higher. For stair ascent, there is a delayed peak acceleration with stair descent and walking having very similar shapes. The difference in peak magnitude between activities is a result of variation in step cadence.

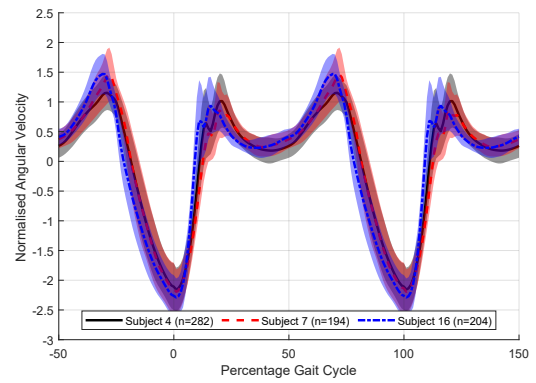
The variations in sensor value between subjects is less than the variations between activities, with early stance having the greatest variations between participants for both sensors plotted. Walking shows the most consistent results among participants. These trends hold true for the subjects not shown.

Simplified LSTM Model Behavior Table 4-3 presents the classification accuracies of each input and class combination. For each model, classification accuracy was recorded for the validation data and a set of unseen test data from excluded participants. It can be seen that all the models performed equally well for both validation and test data sets. Given the simplicity of the models, this suggests that an LSTM can separate activities from only the prominent features for both seen and unseen users, but only to around 80% accuracy.

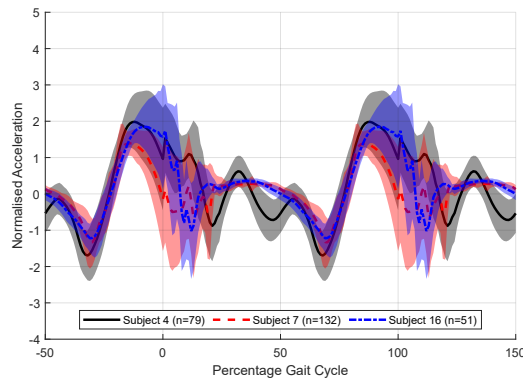
The models struggled to separate stair descent from the other two activities and, apart from



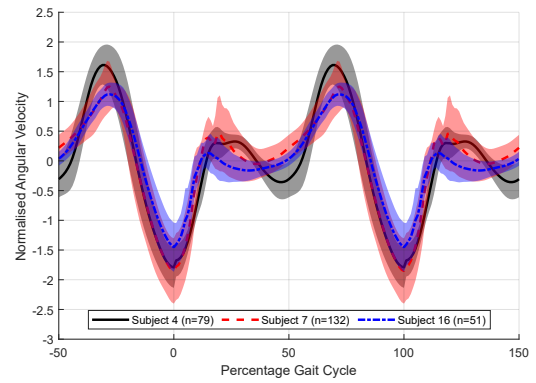
(a) Acceleration in x during W



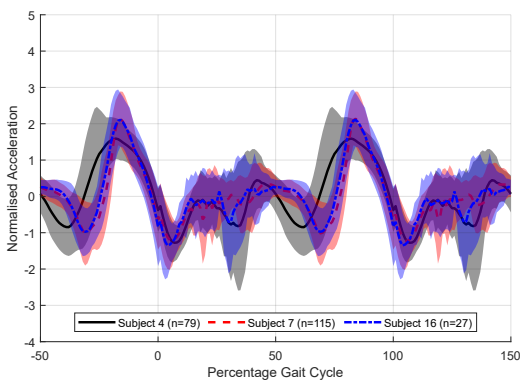
(b) Shank angular velocity about y during W



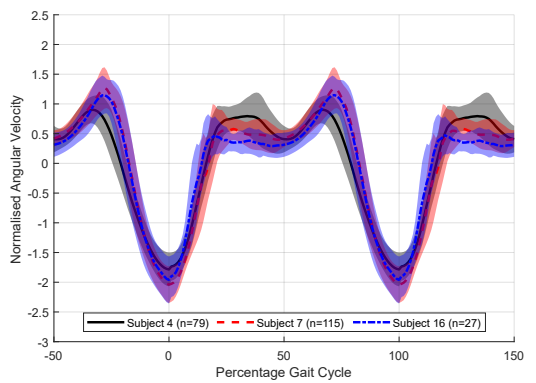
(c) Shank acceleration in x during SA



(d) Shank angular velocity about y axis during SA



(e) Shank acceleration in x during SD



(f) Shank angular velocity about y axis during SD

Figure 4-10: Gait trends for the right shank x accelerometer and y gyroscope for 3 different activities. The solid lines show the mean and the shaded area the standard deviation for n steps. Black solid—Subject 4, Red dashed—Subject 7, Blue dot-dash—Subject 16.

Table 4-3: Summary of simplified model performance.

Model Classes	Sensor	Validation Accuracy	Test Accuracy
W, SA+SD	<i>y</i> Gyroscope	71.6%	86.0%
SA, W+SD	<i>y</i> Gyroscope	82.7%	83.7%
SD, W+SA	<i>y</i> Gyroscope	57.9%	65.6%
W, SA, SD	<i>y</i> Gyroscope	*	*
W, SA+SD	<i>x</i> Accelerometer	86.6%	89.2%
SA, W+SD	<i>x</i> Accelerometer	88.6%	87.1%
SD, W+SA	<i>x</i> Accelerometer	78.4%	81.9%
W, SA, SD	<i>x</i> Accelerometer	71.9%	72.4%
W, SA+SD	<i>x</i> Accel and <i>y</i> Gyro	59.3%	48.8%
SA, W+SD	<i>x</i> Accel and <i>y</i> Gyro	71.2%	67.1%
SD, W+SA	<i>x</i> Accel and <i>y</i> Gyro	75.1%	80.8%
W, SA, SD	<i>x</i> Accel and <i>y</i> Gyro	58.6%	66.2%
W, SA+SD	6 axis IMU	82.0%	83.3%
SA, W+SD	6 axis IMU	74.2%	71.8%
SD, W+SA	6 axis IMU	55.3%	63.3%
W, SA, SD	6 axis IMU	48.0%	50.7%

* Unable to train a model that could classify this set of classes.

with the six-axis IMU, most accurately classified stair ascent. All models performed worst when attempting the hardest task of classifying all three activities individually. An input of the *x* accelerometer on its own performed most accurately, even compared to models with multiple input sensor channels. When using only the *y* gyroscope, it was not possible to separate the three activities individually.

Figures 4-11 and 4-12 show the trends in hidden state for the simplified model at different percentage points through the gait cycle¹. Figure 4-11 has an input of the *y* axis gyroscope and Figure 4-12 the *x* axis accelerometer. In Figure 4-11, the model is classifying stair ascent from a combined class of stair descent and walking. For Figure 4-12, the model is classifying walking from stairs (ascent and descent). Each of the activities is plotted in a different color, solid black for walking, dashed red for stair ascent and dot-dash blue for stair descent. The five subplots show the windows starting at different percentage offset from peak swing. The *x*-axis has units of percentage gait cycle, the *y* axis is the dimensionless output of the hidden state. Values of the *y* axis tend towards -1 , 0 or 1 , depending on the dense layer classifier weights. A value close to these represents a more certain classification.

¹Both figure 4-11 and 4-12 shows the result for subject 04 only. The same result was achieved for all subjects.

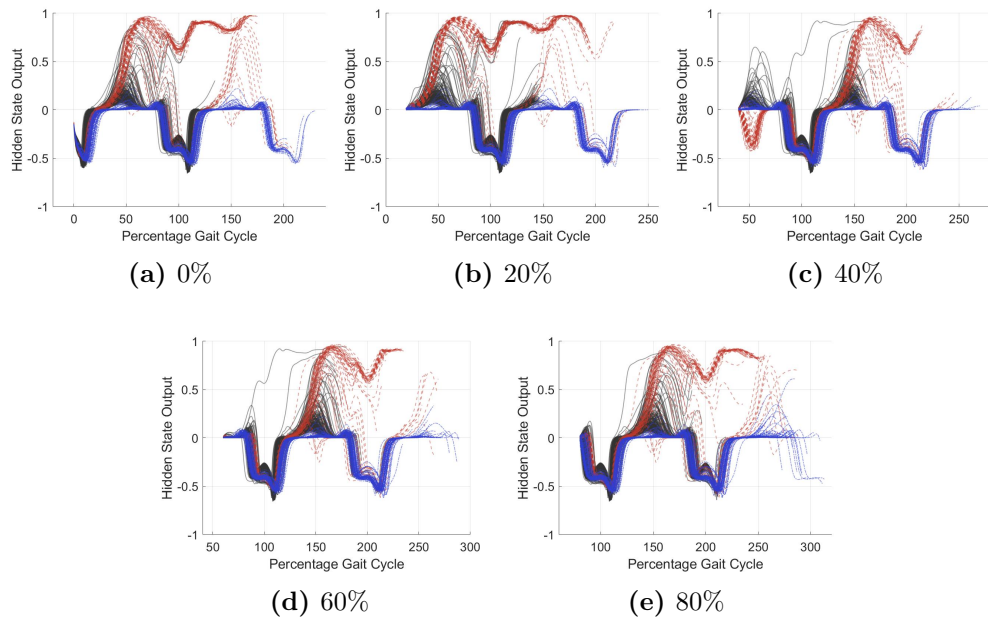


Figure 4-11: Hidden state of single unit LSTM model with x axis accelerometer as its input. The model output classifies stair ascent from walking and stair descent. walking (solid black), stair ascent (dashed red) and stair descent (dot-dash blue). The x axis represent the percentage gait cycle, the y axis is the dimensionless hidden state value, this tends to 1 for stair ascent and 0 for walking and stair descent.

For both acceleration and gyroscope, the hidden state value changes most during early stance. For the y gyroscope, Figure 4-11, it can be seen that the classification of stair ascent from walking and stair descent occurs around 50% gait cycle. For the x accelerometer, Figure 4-12, this occurs later in the gait cycle, around 70%. For the x accelerometer hidden state trends are less tightly grouped, likely due to noisy input data. This may be fixed by input smoothing, further work is required to investigate this. Classification of stair descent is less certain; the model struggles to separate this from the other two classes. Stair ascent and walking are easily classified.

The simplified model is very good at adapting to variation in gait cadence. This can be seen as despite the steps plotted being normalized to gait cycle, the trends in LSTM hidden state were consistent. This suggests there is little need to adjust the input data to account for variations in cadence.

Analysis of the simplified model has demonstrated that even a model of extremely limited learning capacity can achieve reasonable LMR classification accuracy. It has also shown that the classification of activity occurs exclusively within the early stance phase for the three activities examined. This suggests that the model will be highly sensitive in this

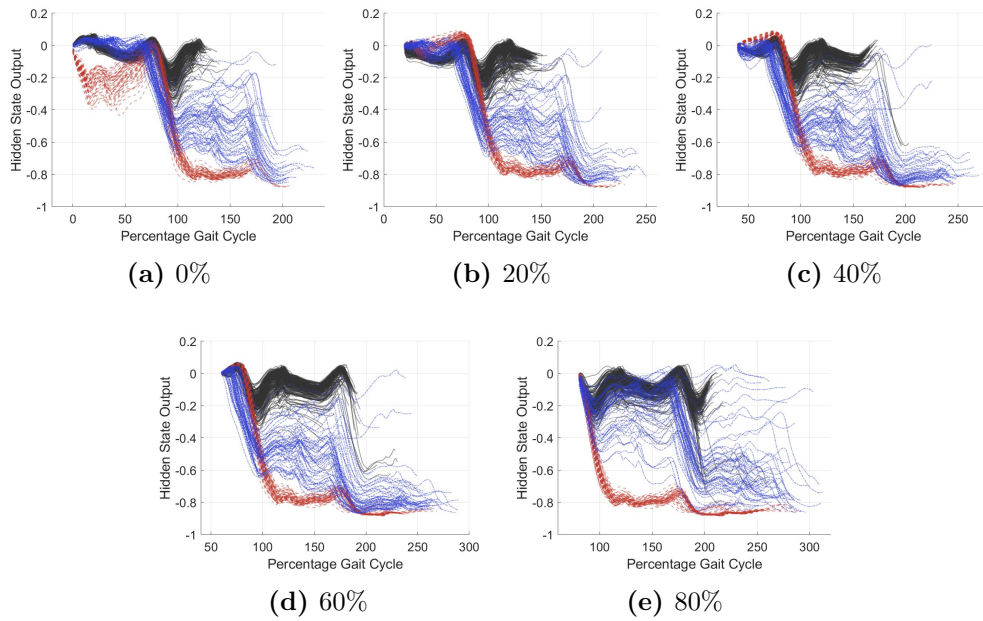


Figure 4-12: Hidden state of single unit LSTM model with x axis accelerometer as its input. The model output classifies walking from stairs (ascent and descent). walking (solid black), stair ascent (dashed red) and stair descent (dot-dash blue). The x axis represent the percentage gait cycle, the y axis is the dimensionless hidden state value, this tends to 0 for walking and -1 for stair ascent and descent.

area. The model also obtained minimal additional information beyond one stance period; therefore, a window of greater than one gait cycle is unnecessary. The learning from this will now be compared to a full complexity model to verify the results broader applicability.

4.2.7 Practical LMR LSTM Network Hyper-Parameter Sensitivities

Within this section, the effect of hyper-parameter selection on model performance for a practical LMR LSTM network will be evaluated. The network architecture used is described in Section 4.2.5.2.

The following hyper-parameters will be investigated: Window size, LSTM units, Number of layers, Different Sensor inputs, and Number of training subjects

Finally, a simple attempt to improve performance around the transition region will be assessed. The transition between activities is highly variable, data label augmentation will be investigated to add a seventh output, transition, to try and identify this area and act as a measure of confidence.

4.2.7.1 Analysis Methodology

For the more complex models, the fully connected links between layers and the hidden state become too convoluted to interpret directly. Instead, classification accuracy will be the primary measure of model performance. This will be given as the percentage of correctly classified windows out of the total input windows. Validation data will be used to evaluate seen data performance, and used test data as a measure of generalization to unseen data.

To investigate the network dimensions, three different window lengths (32, 64 and 128 timesteps at 100 Hz), and six unit widths (4, 6, 8, 16, 32, 64) will be tested. For each model shape, the model was trained five times for the five different train/test data sets. With performance evaluated by classification accuracy.

To evaluate how the number of training subjects effects the performance of the model, models were trained with varying numbers of individuals. Between one and 21 training subjects were tested, with a single subject used as the test set. For each incremental increase in subjects, the model was trained ten times with a different excluded subject.

Miss-classification will be analyzed using confusion matrices. A confusion matrix is a tabular representation of the performance of a classifier. Each cell is populated by a count of the ground truth against the classified output. This allows the accuracy of individual classes and confusion between classes to be assessed.

The time series classification output is also used to identify regions of particular uncertainty. By plotting ground truth and classifying labels as color-coded regions on a time axis, areas of incorrect classification can be assessed.

Finally, the seventh classification output, transition, will be evaluated. To train the model for this, data labels will be augmented with a transition region added for 0.5 s before and after changes in activity. Models of varying hyper-parameters will then be trained with the transition state in and classification accuracy used to evaluate performance.

4.2.7.2 Results and Analysis

Below the results of the experiments on the practical LMR LSTM network are presented and analyzed.

Network Size Figure 4-13 presents the model accuracies achieved for each model \pm standard deviation ($n = 5$). Figures 4-13a,b contain the validation and test classification accuracies, respectively. The 32×4 model contained 1124 parameters, the 128×6 model 67334 parameters.

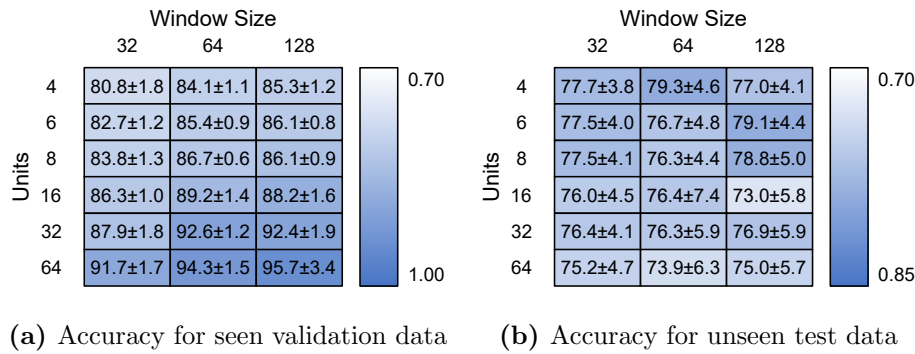


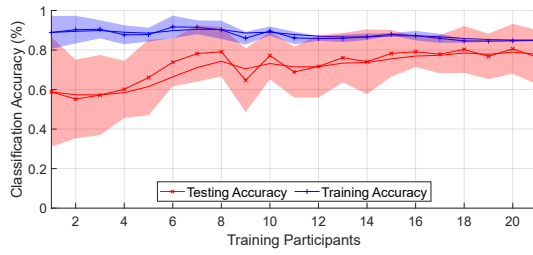
Figure 4-13: Model accuracy for hyper-parameters of different LSTM units and input window size for both seen and novel subjects.

The validation accuracy increases with increasing model size. The improvement when moving from a 32 timestep window to 64 is much greater than when increasing to 128. The number of units was the most direct factor in achieving higher validation accuracy. For test accuracy, the results plateau around 80%, after which the improvements in validation performance likely occur due to over-fitting to individual traits of the training participants. This also corresponds with an increase in standard deviation for the test data set. A complete gait cycle takes approximately one second, 100 timesteps, so it is likely that exceeding this would make little difference. The continuing performance is likely due to the larger, more sophisticated dense layer.

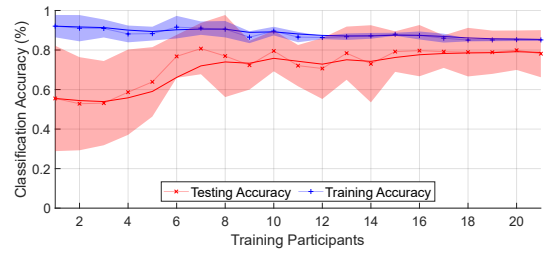
Multi-layer networks were also investigated. Networks with two, three and four deep LSTM layers were tested, but they showed no improvement in generalization and only a small improvement in validation accuracy. The same was observed with multiple sensors; there was no additional improvement in generalization beyond a single 6-axis shank IMU, only an improvement in seen data accuracy. When the three sensor locations were tested individually the shank IMU performed best.

Number of Training Subject Figure 4-14 presents the changes in accuracy for varying numbers of training participants. The red line represents the smoothed unseen test subject classification accuracy average for the ten models trained. The blue line represents the same for the validation data. The solid area represents the standard deviation at each point. Figure 4-14a,b show the results for a 64 timestep 12 unit and 128 time step 6 unit models, respectively.

Figure 4-14 shows that increasing the number of participants leads to better generalized performance; however, the effects on increasing numbers of participants levels off at around 15 participants. This would indicate that for novel subjects to achieve high levels of classi-



(a) 64 Timesteps, 12 Units



(b) 128 Timesteps, 6 Units

Figure 4-14: Classification accuracy of seen/unseen subjects for training with different numbers of participants for two different models.

classification performance, increasing the number of subjects alone may not be enough.

Analysis of Miss-classification Figure 4-15 shows the confusion matrices for a 128 timestep, 6 unit single layer LSTM network. Figure 4-15a is for the training validation data and Table 4-15b the unseen test data. Figure 4-16 shows the same for a model with 128 timesteps and 32 units. The 6 unit model had an overall classification accuracy of 87.4% for validation and 84.7% for test, the 32 unit model accuracy was 96.1% and 76.0%.

	W	RA	RD	SA	SD	S
W	94.1	1.0	1.0	0.9	1.0	2.1
RA	49.3	48.7	1.0	0.3	0.6	0.1
RD	35.4	0.1	62.8	0.0	1.5	0.2
SA	13.8	0.1	0.0	84.7	0.9	0.5
SD	19.2	0.0	0.6	0.4	79.5	0.3
S	19.4	0.1	0.0	0.8	0.5	79.2

(a) Validation

	W	RA	RD	SA	SD	S
W	92.6	0.1	1.0	1.2	2.7	2.4
RA	97.7	0.7	0.8	0.4	0.3	0.2
RD	78.8	0.0	18.0	0.1	2.9	0.2
SA	15.4	0.0	0.0	81.6	1.2	1.7
SD	26.2	0.0	0.7	2.5	70.1	0.5
S	36.5	0.0	0.1	1.1	1.0	61.3

(b) Test

Figure 4-15: 128 timestep, 6 unit confusion matrices.

	W	RA	RD	SA	SD	S
W	97.7	0.5	0.2	0.4	0.3	0.7
RA	11.9	87.2	0.5	0.1	0.2	0.2
RD	6.9	0.1	92.6	0.0	0.2	0.2
SA	4.8	0.1	0.0	94.7	0.3	0.1
SD	4.8	0.0	0.1	0.3	94.7	0.1
S	6.9	0.1	0.1	0.1	0.1	92.7

(a) Validation

	W	RA	RD	SA	SD	S	T
W	84.6	0.3	2.8	1.4	1.5	3.0	6.3
RA	89.3	2.6	0.7	1.3	0.2	0.2	5.6
RD	51.2	0.1	22.7	1.1	1.1	0.2	23.6
SA	13.0	0.1	0.0	66.2	1.2	1.3	18.2
SD	19.7	0.0	1.5	3.1	48.5	1.6	25.7
S	34.8	0.1	0.1	1.0	0.7	60.0	3.2
T	36.5	0.6	1.8	11.9	10.3	2.8	36.2

(b) Test

Figure 4-16: 128 timestep, 32 unit confusion matrices.

It can be seen that nearly all miss-classifications are confusions with walking. The test

data showed a similar pattern for both models, even though the 32 unit model is over-fitted to the training participants. Ramp descent and ascent are both heavily confused with walking. This is likely because of the similarities in gait cycle between the two activities, and the difficulty is accurately labelling this activity due to subject biases. Stairs get slightly confused between each other but again mostly with walking. Stair Descent performs worse than stair ascent. It is not obvious why stop performs poorly, although possibly due to limited data. Some miss-classifications may have occurred due to under labelling or inaccuracies in labelling during recording.

Figure 4-17 shows a visual representation of the activities labelled during a recording, above which is a plot of where the classification errors occurred. As can be seen, a large proportion of miss-classification occur around changes in activity. This is likely because the transition between activities is highly variable and uncertain.

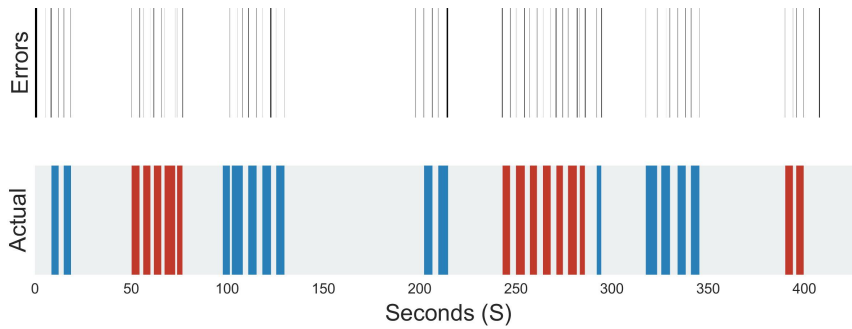


Figure 4-17: Miss-classifications and labelled activity locations, grey—walking, red—stair ascent, blue—stair descent.

Transition State Figures 4-18 and 4-19 present the confusion matrices for the transition models trained with 6 and 32 units, respectively. The 6 unit model achieved 82.8% accuracy on validation data and 72.4% for test data, and the 32 unit model achieved 93.1% and 69.3%. If the transition state is excluded from the classification accuracy then when presented with test data the models achieve 75.2% and 75.0% accuracy for the 6 and 32 unit models, respectively.

The addition of a transition state has not improved classification performance, achieving equal or worse performance than without the transition state, even when excluding the transition state for classification accuracy. This result is unexpected and requires further investigation but at first impression this is due to the transition region being highly uncertain, and so it appears the model cannot map this into a single state. Further work is required to investigate if there are other methods of determining model uncertainty.

	W	RA	RD	SA	SD	S	T
W	94.6	1.1	0.9	0.3	0.5	2.0	0.7
RA	45.6	52.6	1.0	0.2	0.1	0.1	0.5
RD	33.0	0.4	64.3	0.0	0.8	0.3	1.2
SA	10.5	0.1	0.0	83.5	0.9	0.4	4.6
SD	17.4	0.0	0.7	0.4	76.1	0.4	5.0
S	19.1	0.1	0.1	0.5	0.3	78.3	1.8
T	44.1	1.2	2.5	14.9	16.3	2.1	18.8

(a) Training

	W	RA	RD	SA	SD	S	T
W	89.0	0.0	2.1	0.4	2.2	2.5	3.9
RA	96.4	0.1	0.0	0.3	0.2	0.0	3.0
RD	79.3	0.1	13.6	0.0	3.2	0.2	3.7
SA	13.7	0.0	0.1	78.6	0.9	0.3	6.4
SD	23.6	0.0	1.1	1.6	60.2	1.6	11.8
S	28.3	2.0	0.0	0.7	0.2	66.5	2.2
T	47.6	0.2	0.8	15.5	15.6	2.0	18.3

(b) Test

Figure 4-18: 128×6 Transition Model.

	W	RA	RD	SA	SD	S	T
W	97.4	0.4	0.3	0.2	0.2	0.9	0.8
RA	14.7	83.6	0.6	0.1	0.0	0.1	0.9
RD	4.6	0.1	94.1	0.0	0.1	0.2	0.9
SA	4.2	0.1	0.0	91.7	0.7	0.2	3.1
SD	3.7	0.0	0.1	0.5	91.1	0.2	4.5
S	4.6	0.0	0.0	0.2	0.1	94.7	0.4
T	21.7	1.3	2.5	5.9	5.2	1.0	62.4

(a) Training

	W	RA	RD	SA	SD	S	T
W	84.6	0.3	2.8	1.4	1.5	3.0	6.3
RA	89.3	2.6	0.7	1.3	0.2	0.2	5.6
RD	51.2	0.1	22.7	1.1	1.1	0.2	23.6
SA	13.0	0.1	0.0	66.2	1.2	1.3	18.2
SD	19.7	0.0	1.5	3.1	48.5	1.6	25.7
S	34.8	0.1	0.1	1.0	0.7	60.0	3.2
T	36.5	0.6	1.8	11.9	10.3	2.8	36.2

(b) Test

Figure 4-19: 128×32 Transition Model.

4.2.8 Discussion

The study set out to understand the operation of an LSTM LMR network and the effects of its hyper-parameters on classification accuracy and generalization.

Analysis of the simplified model identified that early stance was a prominent feature in the separation of walking from stair ascent and descent, suggesting a high model sensitivity in the early stance region. It was also observed that this was the area of most variation between individuals. Hyper-parameter sets that achieved greater than 80% accuracy reduced the performance of the classifier on unseen data. This suggests that it was over-fitting to individual subject's gait traits reducing generalization. The larger standard deviation in the test set also points to this conclusion. Adjustment of hyper-parameters and standard regularization techniques alone were not sufficient to solve this over-fitting. These two observations may begin to explain the challenges in achieving greater than 80% classification accuracy when presenting novel users to the model. This expands on the observation by Dehghani et al. [210] by suggesting that instead of a 15% reduction in performance with unseen subjects, there is a maximum ceiling of performance of around 80%.

When investigating hyper-parameters, the model was able to demonstrate high levels of performance on the created dataset. Classification performance was comparable to literature achieved similar accuracies to the best performing model. It is noted that due to few

studies being forthcoming on the exact network units, found to be a very important hyper-parameter, direct comparison with literature is challenging.

Prediction of class around transition was challenging, and the addition of a class for this region did not help. Investigation of other methods to solve this is required, such as a form of output averaging or methods for gauging uncertainty from the full classifier output.

Increasing the number of participants in the study did not improve generalization beyond around 80%. It can be theorized that this would only help if the model were trained on a subject with a similar gait to that of the novel user. As amputees have much more varied gait, this approach is unlikely to be practical. There is the potential that a form of data-augmentation may help with this, but this has not been investigated. A more realistic approach is likely a form of individual personalization.

Implementing these techniques in a prosthetic device is still a way off. The results show promise, but further studies are required to address the concerns raised. Practical considerations are also needed; the LSTM model created has a small parameter count, but still requires many calculations to be implemented on an embedded system. Suitable fail safes would also be required to ensure no harm came to the user, especially if it had not been trained to their gait.

The study was limited to non-amputee data to achieve a large enough population. Data from 22 subjects was collected; this is a large data set, but still a relatively small study. It is also still to be determined how applicable the outcomes of this study are to amputees.

The investigation into hyper-parameters was broad, but there are still many more that could have been investigated. Such as the use of IMU sensors from different locations and the filtering of the IMU data.

4.2.9 Conclusions

Within this paper we explored the behavior of an LSTM network trained to complete LMR tasks. In literature, there is a lack of studies investigating the internal operation of an LMR LSTM networks, their hyper-parameter sensitivities and poor novel user generalization.

A new dataset for LMR research of 22 non-amputee subjects performing six activities in a real-world environment was collected. A comparison of sensor data for the gait of different subjects revealed that most variability occurred in early stance. Using the dataset, the behavior of the LSTM layer was examined through mapping input data to changes in hidden state. This revealed that the model primarily classified based on data around early stance.

This behavior could only be directly observed on a simplified model, as the full-connected nature of a practical network makes it too convoluted to interpret hidden state.

A practical LMR LSTM network was trained for a wide variety of hyper-parameter values to determine its sensitivities. Classification accuracy, of both validation and novel users test data, was used to determine the generalization performance. This revealed that although the network can potentially achieve $> 95\%$ accuracy, it is over-fitting to individuals gait traits. This is likely due to the model sensitivities in the most individually variable phase of the gait cycle. There is also an increase in erroneous classification around the transition between activities. None of the hyper-parameters tested were able to account for these issues.

The paper shows that network size, number of individuals training data, and number and location of sensors make insignificant contribution to network generalization performance, demonstrating that personalization is critical.

The outcomes of this work suggested that in order to achieve acceptable accuracy rates (above 95%) for novel users, a form of model personalization is required. Additionally, measures to mitigate the errors around transitions are required. Finally, testing with amputee data is required to determine the applicability of the results to prostheses.

F.S. and P.I. were responsible for the conceptualization of the research. F.S. developed the methodology, software, performed the data curation, investigation, formal analysis and wrote the original draft. A.P. and P.I. supervised the project and reviewed the paper. All authors have read and agreed to the published version of the manuscript.

This research was funded by EPSRC through a studentship administered by the University of Bath; reference EP/N509589/1-1943783.

The study was conducted according to the guidelines of the Declaration of Helsinki, and approved by the REACH Ethics Committee of the University of Bath and received a favourable recommendation. (EP 19/20 003 - 17th Feb 2020).

Informed consent was obtained from all subjects involved in the study.

The data generated during this study is available to download from Zenodo under the Creative Commons Attribution 4.0 license, <https://doi.org/10.5281/zenodo.4390498>

The authors would like to thank Richard Tucker of Amphibian Technologies Limited for loan of the Suunto Movesense devices and their programming equipment.

The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

4.3 Post-commentary

Since being published, seventeen papers have cited the article². A number of these citations have used the paper as an example use case for an LSTM network[214, 215, 216, 217, 218] and data capture methods[115, 180]. The paper was also awarded a Editor’s Choice award. The award is given to articles the journal editors believe are of particularly interesting or important in a field.

Maximum performance for validation data was 96.1% . This classification accuracy is comparable in performance with literature. Uddin et al. achieved 94%[214] and Murad et al. achieved 97%[150]. However the performance of the test set, containing novel unseen individuals, was significantly lower at 76%.

This results supports the conclusion drawn by Dehghani et al. that a substantial drop in performance for unseen subjects is to be expected[210]. This work extends on that conclusion by proposing that the high variability between individuals during early stance is a large source of this error. As when the target user behaviour differs from the training dataset the subject will perform poorly[139].

Therefore for a general model to perform well the training set would need to include individuals with a similar gait pattern to all target subjects. This is highly impractical especially for amputee who have significantly more variability in their gait[45, 46]. It is also harder to get data for amputees due to reduced mobility therefore harder to add them into the general training pool[176].

The paper’s outcomes expressed the need to adapt the trained model to each target individual through a method of personalisation in order to perform adequately. It also noted the need for additional testing to demonstrate the suitability of these techniques for amputees. These needs will be explored further in the subsequent chapters.

²as of the 23rd July 2022

Chapter 5

LMR Model Personalisation

The previous Chapter investigated classification accuracy for a general or subject agnostic LSTM based LMR model. Due to the variability between individuals, greater than 80% classification accuracies could not be achieved for unseen novel subjects. Instead, individual personalisation is necessary to adapt the model to novel subjects.

From the literature review, performed in Chapter 2, gaps for personalisation in LMR following gaps were identified:

- No methods for effectively testing the real-world performance with continuous real-world environments
- There has been little work on understanding how performance improves with increasing target data.

The following naming convention will be used; the subject of personalisation will be referred to as the target, with all other subjects referred to as source.

Within this Chapter, methods for achieving this will be explored. Before attempting to produce a model for amputees, methods will be developed and tested on non-amputees. This Chapter will investigate whether a large population of source data can be used to improve the performance and efficiency of producing a personalised LSTM LMR classifier for a target individual. New methods need to be developed to split up continuous real-world data into representative test and training sets of varying sizes. Techniques for model personalisation techniques that can reduce data requirements and improve model performance are also required.

The Chapter first present methods and materials in Section 5.1. Results of a baseline model trained from only target data are presented in Section 5.2, followed by the results and analysis for personalisation techniques in Sections 5.3 and 5.4. Finally, the discussion and conclusions are presented in Section 5.5.

The contributions of this Chapter are as follows:

- A method for evaluating personalisation LMR models from a set of real world continuous gait data
- Demonstration of the impact on classification performance of increased target training data

5.1 Methods and Materials

Within this section, the methods and materials required to address the research question will be detailed. The section is structured as follows: first, details of an expanded data set of labelled real world HAR data are provided; then, new methods for dividing this data into representative data sets are developed; finally, ML personalisation methods are presented.

5.1.1 Gait Data

A HAR data set, which contains both a large population and a large quantity of data for a small subset, is required for these experiments. A data set containing a large number of subjects has been collected previously. Therefore only additional data for the subset of target subjects is required. These will be Subjects 1, 3 and 9. The additional data was collected the same way as previously; see Section 3.3. Table 5-1 summarises the number of samples and episodes collected for each activity.

Table 5-1: Table of quantities of data samples and episodes collected for each target subject.

	Subject	WALK	RA	RD	SA	SD	STOP
Samples	1	462446	141268	139786	59685	44024	62397
	3	291213	77508	59157	48695	50210	157867
	9	368090	115299	82980	49530	51698	60605
Episodes	1	180	54	44	63	54	53
	3	104	34	23	53	45	27
	9	123	21	27	63	67	35

Table 5-2 contains the demographic data for the three target subjects. The demographic data for the source subject population is include in Appendix D Table D-1.

Table 5-2: Personalisation Target Subject Demographics

	Age	Gender	Height [cm]	Weight [kg]
1	27	Male	185	75
3	25	Male	180	80
9	26	Female	173	65

Only data from the shank-mounted accelerometer and gyroscope will be used. From Chapter 4, the minimal performance improvement was seen for the additional sensors as from previous work.

Data for both the left and right ankle were combined to reduce the data required for the target subject. Equation 5-1 was used to rotate and reflect the left ankle to match the right ankle. In Equation 5-1, V is the original data, and V_t is the transformed data.

Figure 5-1 shows the mean signals from the shank-mounted gyroscope in the sagittal plane for each target subject. Note that the signals shown in the Figure are not normalised. Only stair descent for subject 3 shows any apparent differences between left and right ankles. Therefore it is reasonable to combine ankle data in this way.

$$V_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} V \quad (5-1)$$

5.1.2 Data Division

The HAR data set comprises a series of continuous recordings that may cover multiple different activities and environments. Developing an effective method for dividing this data will be critical to demonstrating the effectiveness of personalisation.

It is highly likely to suffer from poor distribution of classes since activities such as walking are far more prevalent than climbing stairs. ML methods perform best when using balanced data sets; therefore, the poor distribution must be corrected. Additionally, the training and test data sets should ideally not include any data from the same environment so that the test set represents a novel environment. Therefore each unique episode should only be used once across the training and test data sets. Finally, the data division method should allow multiple repeatable unique sets to be constructed for cross-validating performance.

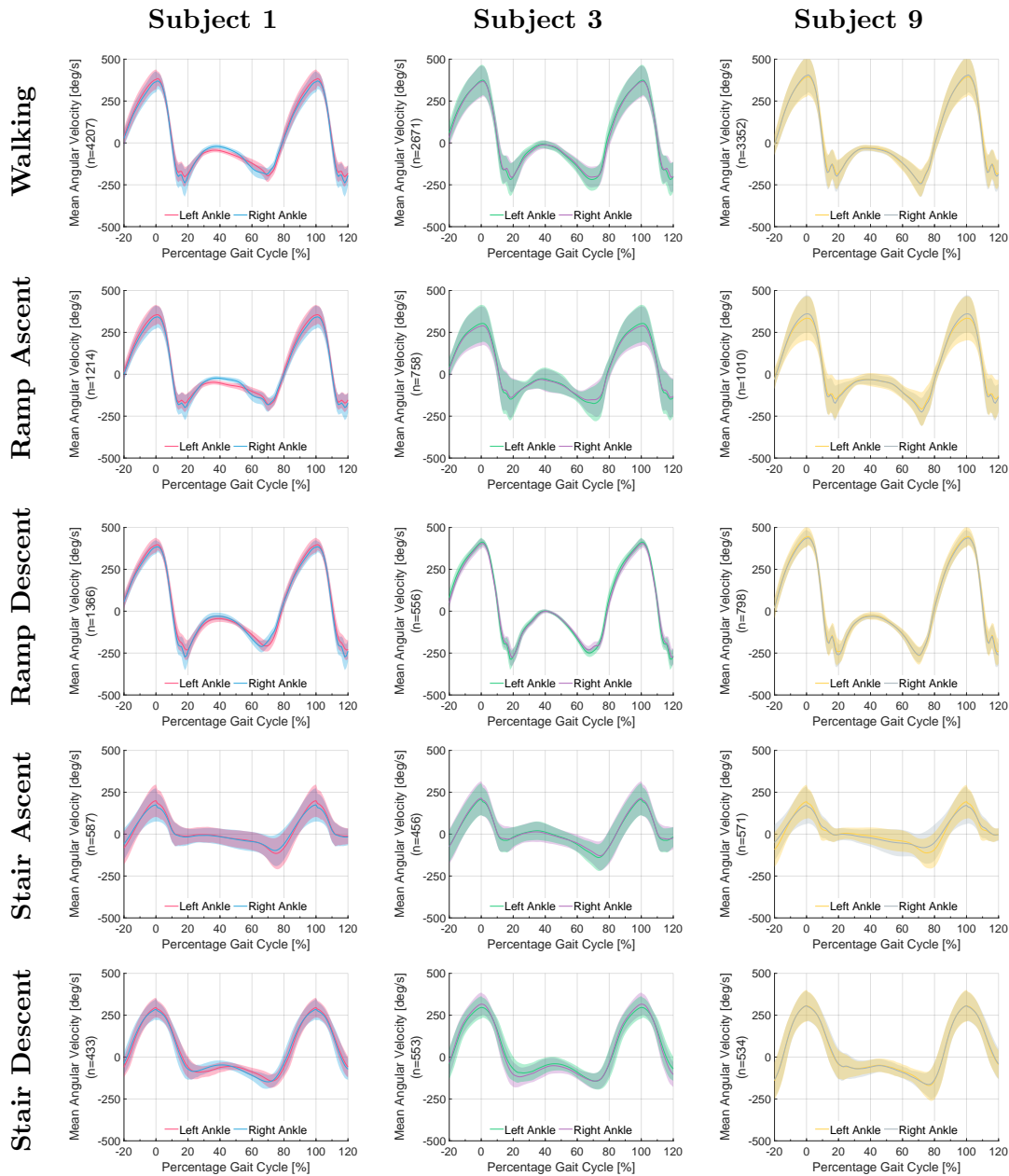


Figure 5-1: The shank’s angular velocity in the sagittal Plane during different activities for each target subject. The solid line shows the mean angular velocity for all steps recorded for each activity. The filled area represents the standard deviation. 0% gait cycle is taken as peak swing for simplicity of calculation. The red, green and yellow lines are for the left ankles of Subjects 1, 3 and 9, respectively. The blue, purple and grey lines show the right ankles of Subjects 1, 3 and 9, respectively.

Achieving all these requirements means that the recordings cannot simply be divided by time.

The proposed approach is to divide the continuous data of each subject into episodes, each containing one continuous period of activity. Episodes can then be combined to form the three independent data sets. The three sets required are training – a set of examples from which the model can learn; validation – used to evaluate the generalisation performance during training; and test – used to evaluate the generalisation performance after training.

Each episode is only used once, with any excess episodes discarded. Excess windows are discarded randomly from all episodes to balance the number labels of each class. To produce cross-validation sets the order of the episodes are shuffled. Figure 5-2 illustrates the process of forming the three data sets.

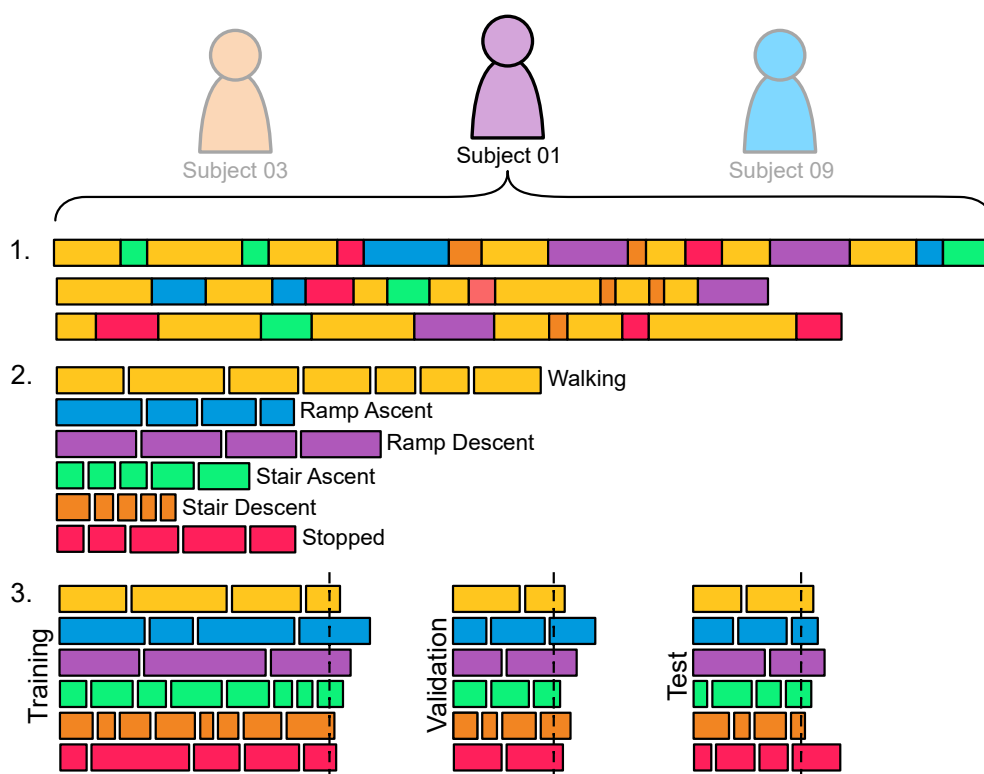


Figure 5-2: Per-episode data division. Step 1 – Labelled data files for a single subject are loaded. Step 2 – Episodes of the same activity are grouped together. Step 3 – Training, Validation and Test sets are formed by stacking episodes until the required window quantity reached.

The test sets will contain 5000 windows of target data for all experiments. The training and validation set will vary in length. The number of training windows will be presented

as the sum of both training and validation windows for conciseness. These will always be in the ratio 70:30.

Each experiment will be repeated multiple times with the episodes included in each set shuffled between each repetition. The shuffling will be repeatable with test data sets drawn first to ensure consistent sets.

The time in seconds can be calculated using Equation 5-2, where T is total set length in seconds, f_s is the sampling frequency, s_k is the window skip value, n_w is the number of windows, l_w is the window size, and n_e is the number of episodes included in the set.

$$T = \frac{1}{f_s}(s_k n_w + l_w n_e) \quad (5-2)$$

Calculating the actual data used in seconds is non-trivial as some windows may be dropped during class balancing. Assuming no windows are dropped, and only one episode is used, 5000 windows uses a minimum of 151 seconds for each class. l_w set to 128, f_s set to 100Hz, and s_k equal to three.

5.1.3 Machine Learning Methods

Two personalisation methods will be evaluated – data supplementation and transfer learning. These will be compared against two baselines; a model trained using only target training data and a general subject agnostic model.

The data supplementation technique will mix source and target data to produce a more extensive training set. This set will then be used to train a new classifier from scratch. The additional data will be selected randomly without attempting to match similar subjects. The amount of both source and target data will be varied to investigate the impact of both.

The transfer learning approach will fine-tune a set of general base models using data from a target subject. The base models will be generated by training from scratch using the complete source data, excluding the three target subjects. Five base models will be produced by randomly shuffling the training and validation.

Personalisation will be performed by additional training using just target data. The amount of target training data used will be varied to assess the impact on classification performance. Three different training configurations will be tested; each configuration will vary by which layers are trained. For the first configuration, all layers will be fine-tuned. The second and third methods will train only the LSTM and dense layers, respectively.

The same LSTM architecture will be used throughout all experiments shown in Figure 5-3. This is the same architecture as used in Chapter 4. The input layer is an LSTM network 128 units. The input to this layer is a 128×6 data window from 6 axis IMU. The LSTM layer uses a hyper-parameter to vary its unit wide but will always be 128 units long. The full output of this layer is then passed to a dense late fusion layer before being passed through a ReLU classifier. TensorFlow allocates 4992 parameters to a 32 unit 128 long LSTM layer and 24582 parameters to the Dense layer.

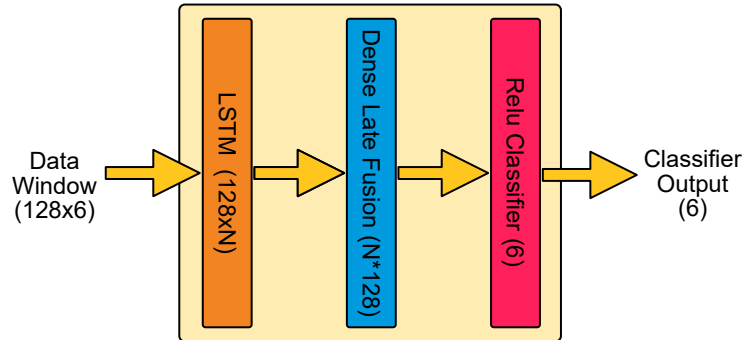


Figure 5-3: Illustration of LSTM machine learning model architecture

All training will be undertaken using the same methods described in Chapter 4. The complete set of windows will be passed through the training systems in mini-batches of 100 windows. After every epoch, the validation set will be used to evaluate the model's performance. Training will be stopped when the categorical loss of the validation set stagnates for more than three epochs. All training hyper-parameters were tuned empirically.

Model performance will be assessed primarily by the classification accuracy using the unseen test data set. Additionally, measures including the number of epochs, training time and quantity of training data required will assess the computationally/data efficiency. By comparison against the baselines, it will be possible to determine if these methods are of benefit.

5.2 Baseline Model Performance

A performance baseline is required to determine if personalisation has resulted in an improvement. Two baselines will be generated for each target subject. These will be the accuracy of a general model for the target and a model trained using only target data. If the performance of the personalisation methods does not exceed the baselines, there is no benefit in them. The performance of both baselines is presented within this section.

For the first baseline, the classification accuracy of the general models when presented with the test data sets was evaluated. The average accuracy of the five models was $75.0\% \pm 2.3$ for Subject 1, $63.9\% \pm 2.9$ for Subject 3 and $77.4\% \pm 5.1$ for Subject 9. The confusion matrices for each subject is presented in Table 5-3. Performance is averaged across the five general models and five test sets. Each cell contains the percentage of total predictions of each class.

Table 1 shows that each target subject struggles in different classes. This is as expected given the likely uniqueness in gait characteristics.

To determine a baseline for models trained with only target training data, LSTM models were trained using increasing amounts of target data. Figure 5-4 shows the classification performance for each subject using different quantities of target data windows for 6, 16, 32 and 64 unit LSTM networks. The complete data tables are available in Appendix A Section A.1.

The maximum performance achieved was 84.4% for Subject 1, 88.5% for Subject 3, and 82.6% for Subject 9. This was achieved at 15000 windows for Subjects 1 and 3 but 9000 samples for Subject 9. It is not clear why performance decreased after this point. Performance of the general model is exceeded at around 1500 windows.

The fastest rate of performance improvement was seen early on, from 100 to 1500 data windows. Beyond this, there was a more gradual increase in performance. It appears that performance would have continued to improve the maximum number of windows tested. Indicating further data would still improve performance.

Standard deviation reduced with increasing data windows, indicating more consistent performance across all test sets as the model was exposed to more data.

The baseline model took on average eight epochs to train with a 95th percentile of 13

In general, increasing the number of units improved classification performance. However, performance levels off at 32 units, indicating diminishing returns beyond this point. Only the 6 unit model appears to have insufficient learning capacity. Increasing the number of units also reduced the number of epochs required to train the models. Therefore 32 units is likely a good candidate for future models.

The reduction in performance at 3000 samples for Subject 3 is likely due to model exposure to a new environment in the training data. Performance recovers with increasing amounts of data. Subject 9 also experiences similar drops in performance.

An assessment of where classification errors occur can be made by looking at the confusion

Table 5-3: Confusion matrices of a general model presented with target subject test data. Columns represent the prediction labels, and the rows represent the real labels. Each value represents the percentage of total predicted labels for that class. Ramp Ascent (RA), Ramp Descent (RD), Stair Ascent (SA), Stair Descent (SD)

(a) Subject 1

		Predicted Classes					
		WALK	RA	RD	SA	SD	STOP
True Classes	WALK	37.9	26.9	24.5	0.1	6.2	6.3
	RA	57.9	65.3	1.2	1.0	0.2	0.0
	RD	2.4	0.9	72.8	0.0	3.7	0.5
	SA	0.5	5.9	0.0	98.6	1.3	2.0
	SD	1.1	0.8	1.5	0.2	88.6	5.8
	STOP	0.2	0.3	0.0	0.0	0.0	85.4

(b) Subject 3

		Predicted Classes					
		WALK	RA	RD	SA	SD	STOP
True Classes	WALK	27.4	29.5	1.7	4.3	5.4	19.6
	RA	40.3	70.2	1.0	0.2	0.1	2.3
	RD	31.6	0.2	94.3	0.5	8.2	0.9
	SA	0.4	0.1	0.3	94.9	0.7	0.9
	SD	0.3	0.0	2.7	0.1	85.6	0.0
	STOP	0.0	0.0	0.0	0.0	0.0	76.3

(c) Subject 9

		Predicted Classes					
		WALK	RA	RD	SA	SD	STOP
True Classes	WALK	59.6	8.5	10.7	0.5	9.3	1.3
	RA	13.3	89.2	0.2	14.6	2.5	1.0
	RD	17.7	0.0	84.3	0.0	15.7	0.0
	SA	4.7	1.7	0.0	78.3	1.2	2.1
	SD	4.7	0.5	4.8	0.5	71.4	2.9
	STOP	0.1	0.0	0.0	6.0	0.0	92.6

Table 5-4: Confusion matrices for a bespoke non-amputee LMR model presented with target subject test data. The 32 unit LSTM model was trained with 15000 target data windows. Columns represent the prediction labels, and the rows represent the real labels. Each value represents the percentage of total predicted labels for that class. (Ramp Ascent (RA), Ramp Descent (RD), Stair Ascent (SA), Stair Descent (SD))

(a) Subject 1

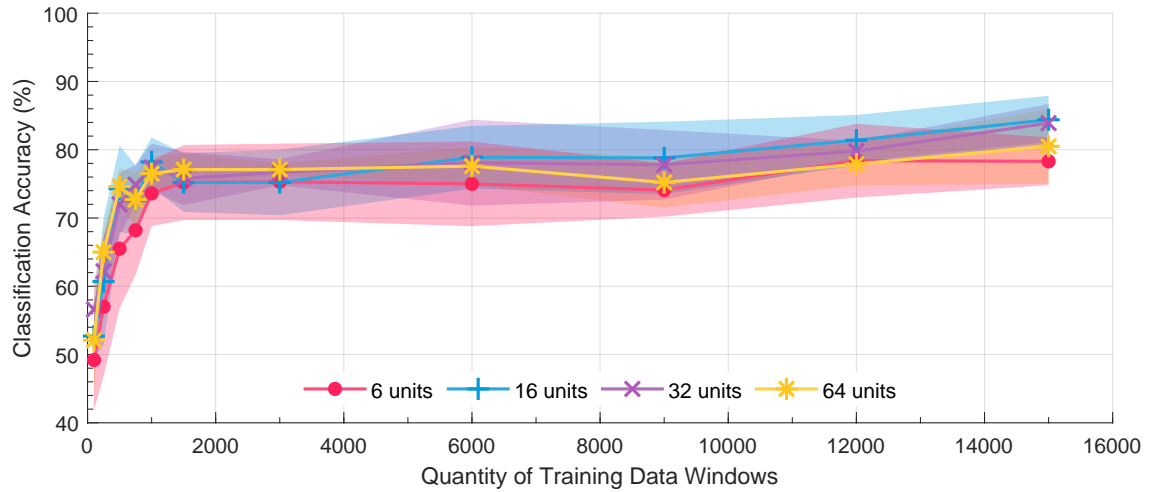
		Predicted Classes					
		WALK	RA	RD	SA	SD	STOP
True Classes	WALK	71.3	20.6	18.1	0.4	1.9	2.7
	RA	25.9	72.4	0.0	0.0	0.0	0.0
	RD	1.2	0.0	80.0	0.2	6.2	0.0
	SA	0.9	5.6	0.2	92.9	5.2	0.2
	SD	0.8	1.2	1.7	4.5	86.7	1.7
	STOP	0.1	0.1	0.0	2.0	0.1	95.4

(b) Subject 3

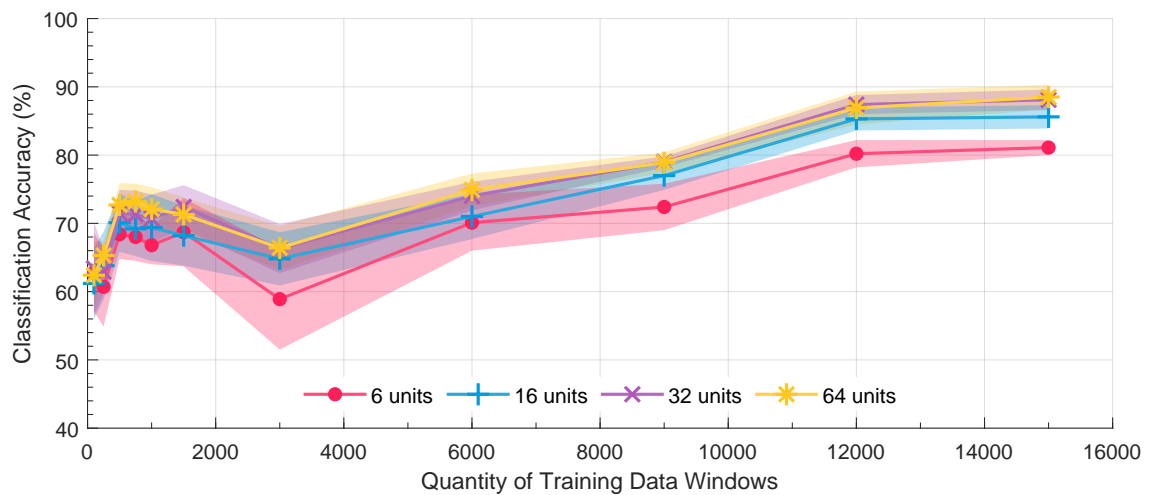
		Predicted Classes					
		WALK	RA	RD	SA	SD	STOP
True Classes	WALK	50.3	3.3	4.2	0.2	4.1	3.1
	RA	32.3	52.1	0.3	0.0	0.0	0.0
	RD	16.5	43.5	92.1	0.1	0.5	0.0
	SA	0.5	0.8	0.2	99.2	1.5	1.0
	SD	0.2	0.2	3.1	0.5	94.0	0.1
	STOP	0.1	0.0	0.0	0.0	0.1	95.7

(c) Subject 9

		Predicted Classes					
		WALK	RA	RD	SA	SD	STOP
True Classes	WALK	90.4	7.6	6.3	2.0	2.7	0.0
	RA	2.2	82.5	0.1	1.4	0.1	0.0
	RD	5.6	7.4	85.1	2.1	8.8	0.0
	SA	1.0	1.7	0.6	92.8	4.7	0.1
	SD	0.8	0.6	8.0	1.6	83.4	0.7
	STOP	0.1	0.2	0.0	0.2	0.4	99.2

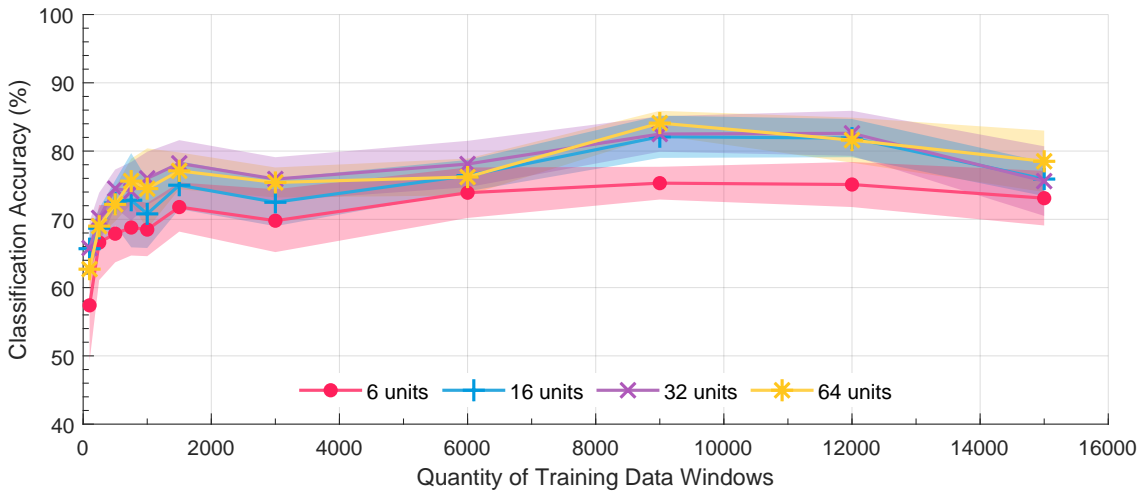


(a) Subject 1



(b) Subject 3

Figure 5-4: Classification performance for different size LSTM networks trained with varying amount of target subject data. The solid lines represent the mean of all models trained; the filled area represents the standard deviation ($n = 10$). Each line shows the classification performance for a different number of LSTM units. The red dot is 6 units, blue plus 16, purple cross 32 and yellow asterisk 64.



(c) Subject 9

Figure 5-4: Classification performance of different size LSTM networks trained with varying amount of target subject data (Cont.).

matrices. Table 5-4 shows confusion matrices for the three targets classifiers created using 15000 training windows. Performance is averaged across the five bespoke models test sets. Each cell contains the percentage of total predictions of each class.

The confusion matrices show that the stop class achieves the highest accuracy, greater than 95% for all subjects. As this is a very distinct class, this should be expected. Stairs were also identified reasonably accurately, with Stair Ascent achieving greater than 92% accuracy and Stair Descent greater than 83%. The classifier struggled to distinguish walking from Ramp Ascent and Ramp Descent for all subjects with accuracy as low as 50%.

Comparisons between the two confusion matrices, Tables 5-3 and 5-4 show that each perform better in different classes. Therefore, combining the knowledge from both data sources should improve performance.

5.3 Data Supplementation

The first personalised model technique investigated will be data supplementation. This involves supplementing target data with a varying amount of data from a general source set to form a larger training set. The source data is made up of a larger number of non-amputee subjects. The experiment consists of mixing 100 to 3000 windows of target data with between 100 and 3000 source data windows.

A series of LSTM models were trained using different quantities of source and target win-

dows. The experiment aimed to establish if the addition of source data improves classification performance.

Tables 5-5, 5-6 and 5-7 present the results of all experiments. Each cell contains the mean classification accuracy for target training and standard deviation. Columns represent different quantities of source training windows. Table rows represent different quantities of target training windows. The highest classification accuracy for each quantity of target training windows has been highlighted in bold.

Except for low source data quantities with high quantities of target data, supplemental source data improves classification performance over the baselines.

There are no apparent trends for how much additional data is required. More source windows than target windows always increase classification accuracy. Classification performance increases with increasing source data before falling off. There is no obvious point where degradation begins occurs. This may be due to the random nature of the selected source windows.

This method requires extensive training resources. Both a large amount of training data must be used and training takes a large number of epochs. The average number of epochs was 21. with a 95th percentile of 38 epochs.

This method looks challenging to implement successfully as there it is difficult to predict where the best performance occurs. It also requires a lot of computation resources to train each model, so empirically determining this point is computationally expensive.

The F1-scores for each class show that SA, SD and S all performed extremely well. Each of these three classes achieving an F1-score above 0.91. RA and RD performed worse achieving F1-scores of 0.70 and 0.82 respectively. Walking performed significantly worse than the other five classes with an F1-score of 0.45. This suggest that walking was frequently miss-classified with ramp ascent and descent.

5.4 Transfer Learning

Transfer learning involves using the knowledge captured in an existing model as a starting point to building a personalised model. The five general models produced previously were used as the starting point for this experiment. Varying quantities of target subject data windows were then used to fine-tune the target models. By freezing the different network layers, attempts to reduce the computation load required to train the model could be made.

Classifiers were trained for each of the three target subjects by fine-tuning the general models

Table 5-5: Table of classification accuracy for Subject 01 for a model trained using varying amounts of Source and Target training data. The cell value represents the percentage classification accuracy $\pm\sigma$ ($n = 10$). The highest classification accuracy for each quantity of target windows has been highlighted in bold.

		Source Training Windows						
		0	100	250	500	750	1000	1500
Target Training Windows	100	0.566 \pm 0.04	0.759 \pm 0.03	0.785 \pm 0.04	0.787 \pm 0.03	0.803 \pm 0.03	0.803 \pm 0.02	0.800 \pm 0.04
	250	0.622 \pm 0.05	0.773 \pm 0.03	0.816 \pm 0.03	0.828 \pm 0.03	0.836 \pm 0.02	0.817 \pm 0.03	0.839\pm0.03
	500	0.720 \pm 0.05	0.779 \pm 0.02	0.804 \pm 0.02	0.803 \pm 0.02	0.828 \pm 0.03	0.838\pm0.02	0.836 \pm 0.02
	750	0.749 \pm 0.03	0.791 \pm 0.02	0.790 \pm 0.02	0.799 \pm 0.02	0.806 \pm 0.01	0.828 \pm 0.02	0.828 \pm 0.02
	1000	0.774 \pm 0.04	0.791 \pm 0.04	0.807 \pm 0.03	0.796 \pm 0.02	0.806 \pm 0.03	0.821 \pm 0.03	0.833 \pm 0.03
	1500	0.758 \pm 0.04	0.786 \pm 0.02	0.780 \pm 0.02	0.793 \pm 0.03	0.802 \pm 0.03	0.798 \pm 0.03	0.814 \pm 0.02
	3000	0.767 \pm 0.02	0.748 \pm 0.06	0.747 \pm 0.04	0.786 \pm 0.04	0.787 \pm 0.03	0.775 \pm 0.04	0.790 \pm 0.04
	6000	0.781 \pm 0.06	0.784 \pm 0.04	0.784 \pm 0.04	0.781 \pm 0.04	0.787 \pm 0.03	0.782 \pm 0.04	0.800 \pm 0.05
	9000	0.778 \pm 0.05	0.754 \pm 0.03	0.785 \pm 0.03	0.758 \pm 0.05	0.784 \pm 0.04	0.776 \pm 0.04	0.755 \pm 0.05
	15000	0.798 \pm 0.01	0.806 \pm 0.04	0.774 \pm 0.05	0.778 \pm 0.03	0.773 \pm 0.04	0.810 \pm 0.03	0.780 \pm 0.05

		Source Training Windows				
		3000	6000	9000	12000	15000
Target Training Windows	100	0.820\pm0.03	0.806 \pm 0.02	0.804 \pm 0.03	0.789 \pm 0.04	0.809 \pm 0.04
	250	0.815 \pm 0.04	0.827 \pm 0.04	0.831 \pm 0.03	0.801 \pm 0.04	0.812 \pm 0.03
	500	0.825 \pm 0.03	0.826 \pm 0.04	0.832 \pm 0.03	0.824 \pm 0.04	0.819 \pm 0.04
	750	0.854\pm0.02	0.821 \pm 0.04	0.835 \pm 0.03	0.813 \pm 0.03	0.826 \pm 0.03
	1000	0.829 \pm 0.03	0.834 \pm 0.03	0.838\pm0.04	0.821 \pm 0.04	0.834 \pm 0.04
	1500	0.828 \pm 0.03	0.848 \pm 0.03	0.854\pm0.03	0.829 \pm 0.04	0.834 \pm 0.04
	3000	0.817 \pm 0.02	0.844\pm0.02	0.833 \pm 0.03	0.826 \pm 0.02	0.831 \pm 0.03
	6000	0.797 \pm 0.03	0.815 \pm 0.02	0.824 \pm 0.04	0.825 \pm 0.03	0.830\pm0.03
	9000	0.791 \pm 0.04	0.816 \pm 0.04	0.797 \pm 0.03	0.824\pm0.04	0.804 \pm 0.04
	15000	0.810 \pm 0.04	0.819 \pm 0.03	0.817 \pm 0.04	0.830 \pm 0.02	0.833\pm0.04

Table 5-6: Table of classification accuracy for Subject 03 for a model trained using varying amounts of Source and Target training data. The cell value represents the percentage classification accuracy $\pm\sigma$ ($n = 10$). The highest classification accuracy for each quantity of target windows has been highlighted in bold.

		Source Training Windows						
		0	100	250	500	750	1000	1500
Target Training Windows	100	0.634 \pm 0.07	0.791 \pm 0.02	0.817 \pm 0.03	0.844 \pm 0.01	0.862\pm0.02	0.850 \pm 0.02	0.856 \pm 0.02
	250	0.629 \pm 0.03	0.794 \pm 0.03	0.831 \pm 0.02	0.846 \pm 0.02	0.859 \pm 0.01	0.857 \pm 0.02	0.851 \pm 0.02
	500	0.715 \pm 0.03	0.813 \pm 0.02	0.836 \pm 0.03	0.861 \pm 0.01	0.853 \pm 0.03	0.845 \pm 0.03	0.859 \pm 0.02
	750	0.713 \pm 0.03	0.810 \pm 0.03	0.841 \pm 0.01	0.862 \pm 0.01	0.846 \pm 0.02	0.852 \pm 0.02	0.859 \pm 0.03
	1000	0.707 \pm 0.03	0.791 \pm 0.03	0.838 \pm 0.01	0.857 \pm 0.02	0.856 \pm 0.02	0.840 \pm 0.03	0.868 \pm 0.03
	1500	0.724 \pm 0.03	0.791 \pm 0.01	0.813 \pm 0.02	0.830 \pm 0.02	0.853 \pm 0.01	0.866 \pm 0.01	0.870 \pm 0.02
	3000	0.663 \pm 0.04	0.692 \pm 0.04	0.726 \pm 0.03	0.739 \pm 0.04	0.770 \pm 0.04	0.754 \pm 0.05	0.822 \pm 0.03
	6000	0.740 \pm 0.02	0.754 \pm 0.03	0.782 \pm 0.02	0.776 \pm 0.03	0.792 \pm 0.02	0.803 \pm 0.02	0.803 \pm 0.04
	9000	0.790 \pm 0.01	0.791 \pm 0.02	0.791 \pm 0.02	0.802 \pm 0.01	0.805 \pm 0.02	0.814 \pm 0.03	0.820 \pm 0.02
	15000	0.874 \pm 0.01	0.871 \pm 0.02	0.866 \pm 0.01	0.871 \pm 0.02	0.872 \pm 0.01	0.876 \pm 0.02	0.885 \pm 0.01

		Source Training Windows				
		3000	6000	9000	12000	15000
Target Training Windows	100	0.837 \pm 0.03	0.854 \pm 0.04	0.837 \pm 0.03	0.847 \pm 0.03	0.842 \pm 0.02
	250	0.850 \pm 0.02	0.842 \pm 0.04	0.862\pm0.03	0.861 \pm 0.03	0.859 \pm 0.02
	500	0.863 \pm 0.02	0.867\pm0.02	0.840 \pm 0.03	0.849 \pm 0.03	0.860 \pm 0.03
	750	0.860 \pm 0.02	0.874\pm0.02	0.861 \pm 0.02	0.862 \pm 0.02	0.855 \pm 0.01
	1000	0.869 \pm 0.03	0.867 \pm 0.03	0.876\pm0.03	0.871 \pm 0.02	0.866 \pm 0.02
	1500	0.868 \pm 0.02	0.880\pm0.03	0.877 \pm 0.02	0.875 \pm 0.02	0.880\pm0.02
	3000	0.853 \pm 0.03	0.871 \pm 0.02	0.871 \pm 0.03	0.884\pm0.02	0.872 \pm 0.03
	6000	0.837 \pm 0.02	0.863 \pm 0.01	0.873\pm0.01	0.871 \pm 0.01	0.861 \pm 0.02
	9000	0.850 \pm 0.02	0.869 \pm 0.01	0.873\pm0.01	0.869 \pm 0.01	0.872 \pm 0.01
	15000	0.884 \pm 0.02	0.899 \pm 0.01	0.894 \pm 0.01	0.897 \pm 0.01	0.903\pm0.01

Table 5-7: Table of classification accuracy for Subject 09 for a model trained using varying amounts of Source and Target training data. The cell value represents the percentage classification accuracy $\pm\sigma$ ($n = 10$). The highest classification accuracy has been highlighted in bold.

		Source Training Windows						
		0	100	250	500	750	1000	1500
Target Training Windows	100	0.658 \pm 0.04	0.762 \pm 0.04	0.771 \pm 0.04	0.787\pm0.03	0.780 \pm 0.03	0.787\pm0.05	0.787\pm0.04
	250	0.702 \pm 0.04	0.813 \pm 0.02	0.831 \pm 0.04	0.834 \pm 0.03	0.845\pm0.03	0.837 \pm 0.02	0.794 \pm 0.04
	500	0.745 \pm 0.03	0.825 \pm 0.03	0.832 \pm 0.02	0.841 \pm 0.02	0.842 \pm 0.02	0.843 \pm 0.03	0.846 \pm 0.03
	750	0.741 \pm 0.04	0.817 \pm 0.03	0.831 \pm 0.03	0.848 \pm 0.02	0.848 \pm 0.02	0.844 \pm 0.03	0.849\pm0.03
	1000	0.760 \pm 0.04	0.826 \pm 0.03	0.851 \pm 0.03	0.852 \pm 0.03	0.855 \pm 0.02	0.844 \pm 0.03	0.862\pm0.02
	1500	0.782 \pm 0.03	0.807 \pm 0.03	0.827 \pm 0.03	0.847 \pm 0.02	0.828 \pm 0.04	0.849 \pm 0.03	0.854 \pm 0.03
	3000	0.759 \pm 0.03	0.756 \pm 0.03	0.775 \pm 0.03	0.788\pm0.04	0.769 \pm 0.03	0.768 \pm 0.04	0.772 \pm 0.04
	6000	0.781 \pm 0.03	0.788 \pm 0.02	0.770 \pm 0.03	0.778 \pm 0.03	0.794 \pm 0.03	0.796 \pm 0.05	0.775 \pm 0.03
	9000	0.825 \pm 0.03	0.827 \pm 0.03	0.834 \pm 0.02	0.825 \pm 0.02	0.829 \pm 0.02	0.813 \pm 0.03	0.854 \pm 0.02
	15000	0.826 \pm 0.03	0.825 \pm 0.02	0.832 \pm 0.02	0.825 \pm 0.03	0.815 \pm 0.02	0.818 \pm 0.03	0.805 \pm 0.03

		Source Training Windows				
		3000	6000	9000	12000	15000
Target Training Windows	100	0.735 \pm 0.05	0.711 \pm 0.06	0.700 \pm 0.03	0.710 \pm 0.04	0.715 \pm 0.03
	250	0.795 \pm 0.04	0.771 \pm 0.05	0.765 \pm 0.05	0.757 \pm 0.05	0.749 \pm 0.04
	500	0.855\pm0.03	0.801 \pm 0.05	0.795 \pm 0.02	0.798 \pm 0.05	0.794 \pm 0.03
	750	0.849\pm0.03	0.832 \pm 0.03	0.814 \pm 0.04	0.800 \pm 0.05	0.815 \pm 0.03
	1000	0.845 \pm 0.03	0.828 \pm 0.04	0.816 \pm 0.04	0.818 \pm 0.03	0.820 \pm 0.04
	1500	0.838 \pm 0.03	0.847 \pm 0.03	0.850 \pm 0.04	0.820 \pm 0.05	0.861\pm0.02
	3000	0.732 \pm 0.04	0.762 \pm 0.05	0.777 \pm 0.05	0.771 \pm 0.04	0.781 \pm 0.03
	6000	0.796 \pm 0.04	0.812 \pm 0.04	0.817\pm0.04	0.815 \pm 0.03	0.811 \pm 0.03
	9000	0.827 \pm 0.03	0.833 \pm 0.03	0.831 \pm 0.03	0.858\pm0.02	0.827 \pm 0.03
	15000	0.809 \pm 0.04	0.832 \pm 0.03	0.822 \pm 0.05	0.834 \pm 0.03	0.861\pm0.02

produced earlier. Figure 1 shows the classification performance for the three methods of fine-tuning the general models with increasing target training windows. The three methods are fine-tuning the whole network, only fine-tuning the dense layer and only fine-tuning the LSTM layer.

Figure 5-5a shows the classification performance for fine-tuning all layers. The highest classification accuracy observed for each subject was 85.6%, 90.5% and 84.7% for subjects 1, 3 and 9, respectively. This is a significant improvement over the baseline model performances. These results were all achieved with either 12000 or 15000 target training windows, these are at the higher end of the available data. All models completed training in a small number of epochs, on average 5 with a 95th percentile of 8.

Classification performance improves rapidly up to 1500 target training windows. When performance for 100, 250 and 500 training windows is compared to the bespoke baseline model, performance is over 10% better for all subjects. This potentially means only 30 seconds of each class are required. Performance improvements then increase more slowly with additional target data. The improvement over the bespoke model baseline also narrows to only a couple of percent.

Target windows quantities of 100-200 for subject 3 are below the baseline performance. For subjects 3 and 9, performance drops down around 3000 target samples. Other than these values, all models exceed the baseline performance.

The drop in performance for higher values of training windows is still an improvement over the baselines for Subject 9; however, it briefly drops below the general baseline model for Subject 3. It is not apparent why this decrease in performance occurs but could be due to introducing a significantly different previously unseen environment to the training set.

Fine-tuning only the dense layer improved the performance of Subject 3, no improvement for 9 and reduction in performance for 1. Standard deviation in general increased for all three subjects showing more significant variation in performance across all models. When the dense layer was frozen, there was a slight improvement for subject 1 of greater than 1% across all target window quantities but no meaningful change for subjects 3 and 9. Standard deviation remained largely the same when compared to fine-tuning all layers.

This approach to personalisation appears to give significant improvements in performance using only a small amount of target data. Only fine-tuning some layers did not give any consistent or meaningful improvement in performance.

The F1-scores for the different activities were closer for the transfer learning approach compared with data supplementation. Walking again has the lowest F1 score with 0.748.

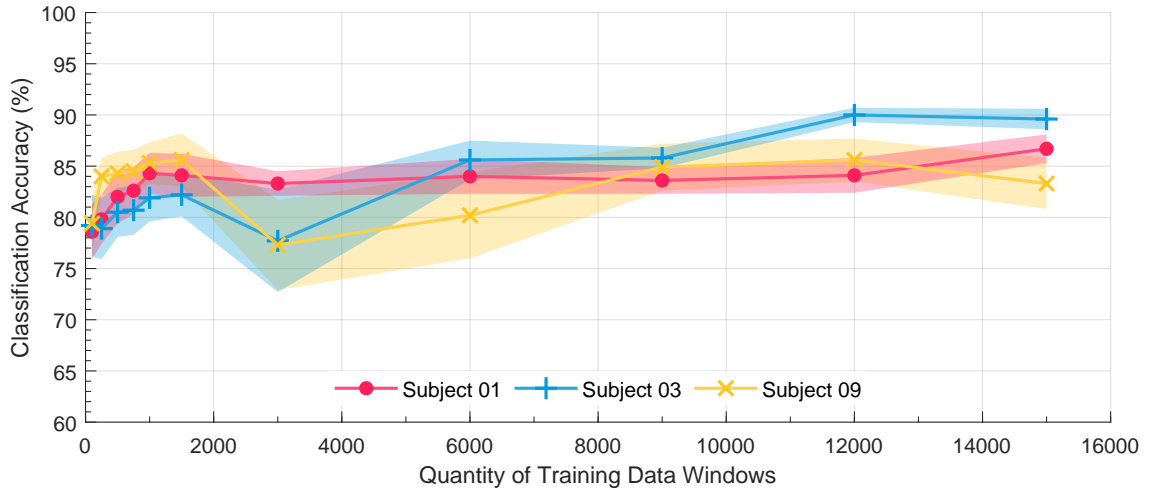


(a) Fine-tuning all layers



(b) Fine-tuning only the dense layer

Figure 5-5: Results of fine-tuning a generic 32 unit LSTM model using increasing amounts of target data. The solid line represents the mean classification performance for each amount of training window. The filled area represents the standard deviation ($n = 25$). Each of the three target subjects is represented individually. The red dot line is Subject 1; the blue plus is subject 3; the yellow cross is subject 9.



(c) Fine-tuning only the LSTM layer

Figure 5-5: Results of fine-tuning a generic 32 unit LSTM model using increasing amounts of target data (Cont.).

Stair descent has the highest score with 0.903 on average. The other four categories fell in-between these two. This shows a much more balanced learning between the classes than the data supplementation.

5.5 Discussion and Conclusions

This Chapter aimed to develop methods that improve the performance of an LMR network for a specific previously unseen individual. Improvements were measured in both computation efficiency and classification accuracy.

First, a real-world representative test scheme was developed. This presented the trained model with an episode of activity that it had not previously seen. It gave a challenging data set as test performance was significantly below training performance. Given that the actual activity environment for each episode is not known, it is difficult to assess how realistic this testing scheme is.

Compared to the previous Chapter, the data division scheme employed also reduced the transition data that the model encountered as the data division scheme cleaned it up. Previous work indicated this is a significant source of inaccuracy that has not been considered.

Two baselines were generated to compare performance against: a set of general models and models trained with only a target subjects data. These established a minimum performance that must be achieved for the personalisation techniques to be of merit.

Two methods were demonstrated, supplementing target data to form an enlarged training set and fine-tuning a pre-trained general model. Both methods successfully used the source data set to improve the classification accuracy for the target users over the baseline. After fine-tuning, subject 3 achieved a maximum classification accuracy of 90.5% an improvement over 26.6% over the general baseline model. Across all techniques and subjects, improvement in accuracy varied from greater than 10% to just over 2% over the baseline.

Both methods were successful in improving performance through personalisation. They also reduced the target data required for the same classification as the bespoke target models. However, the data supplementation method required a significantly greater number of epochs than transfer learning to achieve this performance. For data supplementation, it was challenging to predict how much source data was needed, so would implement in practice would be difficult. These results suggest that transfer learning is a better approach than data supplementation.

All experiments investigated how much target data is required to achieve good performance. The results demonstrated that more data is better, as should be expected. A plateauing of performance with increasing data was not seen, implying additional data would likely have continued to improve performance. Performance improved rapidly early on; therefore, these methods offer the best cost-benefit at lower data quantities.

Collecting additional target data beyond what was collected rapidly becomes unrealistic. It is impossible to expose the model to every possible environment before deployment. A form of continuous online learning may resolve this issue and allow adaption to changes in individual gait characteristics over time.

For lower quantities of target data, data supplementation achieved better results than transfer learning. However, due to the difficulties in predicting how much source data is required, this is difficult to implement. The imprecision of this technique is potentially an issue for this. More precision in selecting which supplemental source data to use, as shown in literature, may improve performance. However, additional measures would be needed to evaluate the similarity between subjects, which is unnecessary when using a purely deep learning approach. An alternative approach could be to bias learning towards the targets data, as shown by Ferrari et al.[164].

Transfer learning performance was more consistent in its improvement in performance, performing better than baseline for the vast majority of tested configurations. The anomalous results appear to come from the training sets used as they were repeated throughout all experiments. This may well be due to the introduction of new environments that present different gait characteristics from those previously seen in the data set. Due to the black-

box nature of each data recording, it is not possible to know which kind of environment each episode is taken from. Therefore, experimentation to determine if this hypothesis is true is impossible without collecting additional data with greater ground truth or controls.

The freezing of layers in the transfer learning did not result in any significant reduction in classification accuracy. For a model as small as was tested, neither did it significantly improve computational performance. For more complex architectures, however, this could be a valuable technique for reducing computational training requirements, as demonstrated by Yoon et al.[159].

The advantages of the personalisation approach demonstrated is that a higher performing model can be produced without as much data and using less training resource for the individual. However this is depended on the general population used as a starting point containing individuals of close gait characteristics to the target subject. If this is not the case then the performance benefits may be severely limited.

Additional areas of research that could improve performance include the investigation of combined data supplementation and transfer learning methods; Additional investigation of model hyper-parameter and different model architectures.

The methods developed in this study will now be taken forward and applied to amputee data to determine if they are still applicable to subjects with abnormal gaits.

Chapter 6

Applicability to Amputee Data

The collection of labelled training data is arduous; this problem becomes more acute when the subject has restricted movement, such as an amputee. Therefore, any system that can reduce these individuals' data requirements is of benefit. In Chapter 5, methods for personalisation of a machine learning model using additional data from other subjects were demonstrated to improve the model's performance and reduce the data requirements for that model. Within this Chapter, these methods will be implemented for an amputee to investigate if they apply to someone with an impaired gait.

The contributions of this Chapter are:

- Collection of amputee gait data that is directly comparable to non-amputee data
- Comparison of shank MARG data between non-amputee, intact limb and prosthetic
- Demonstration of performance differences of LMR network of intact and prosthetic side
- Demonstration of transfer learning from non-amputee data to an amputee for MARG gait data

First Section 6.1 explains the methods used, and presents the collected amputee data. Following this in Sections 6.2, 6.3 and 6.4 present the results of a baseline model performance and performance of two personalisation methods, respectively. Finally, the discussion and conclusion are given in Section 6.5.

From the literature the following gaps were identified:

- No work has attempted to use transfer learning to personalise a non-amputee LMR model for an amputee
- No papers have investigated the difference in classification performance between amputated and in-tact limb

The remainder of this Chapter will focus on investigating these gaps.

6.1 Methods and Materials

The methods used in this Chapter will mirror those used in previous chapters. Additional data for an amputee was collected. This was used to generate a new set of baselines to compare against and implement the previously developed personalisation methods. Complete tables of results for all experiments can be found in Appendix A Section A.3.

Data was collected from a single left trans-tibial amputee wearing a Blatchford Echelon VT prosthetic limb. Blatchford Product Limited collected the data. The data was collected in the same manner as previously described; however, a clinician at the centre held the smartphone to annotate activities to reduce the fall risk for the amputee. Data was collected walking around Blatchford’s site in Basingstoke, both inside and outside. Table 6-1 summarises the data collected, including the number of data samples and the number of episodes for each activity.

Table 6-1: Summary of amputee gait data collected

Activity	WALK	RA	RD	SA	SD	STOP
Samples	38114	6159	7194	2872	2450	11763
Episodes	26	7	7	4	4	15

As significantly less data is available to test with, adjustments had to be made to the quantities used in each data set. The test data set was reduced from 5000 windows to 250 windows. The range of training windows tested was reduced to between 100 and 750. Additionally, to ensure that there were sufficient episodes for SA and SD, each of the stair episodes was split up into new episodes containing a maximum of 200 samples each. This increased the amount of available episodes for both stair activities.

Due to asymmetry, the left and right ankle data cannot be combined. This will be described further in the following section. All other quantities and hyper-parameters remained the same.

6.1.1 Amputee Gait Data

Gait asymmetry of amputees means that, unlike in the previous Chapter, the left and right ankle data cannot be combined to increase the dataset size. Instead, both sides will be evaluated independently to investigate the performance differences between the side. Figure 6-1 shows the angular velocity of the shank in the sagittal plane for the intact and prosthetic limbs of a left amputated trans-tibial amputee. The left ankle has been transformed, so both ankles are presented in the same axes. For comparison, the ankle angular velocity of the non-amputee subject one is shown.

From a visual assessment of the plots, differences between the non-amputee, intact and amputated limbs can be seen. The prosthetic limb shows significant differences to both the intact and non-amputee. The non-amputee and intact limb are closer in appearance.

Differences in the prosthetic limb are especially prevalent during heel strikes (at approximately 20% of the gait cycle), where a much lower angular velocity is observed. In general, a more significant standard deviation is also seen for the prosthetic limb, suggesting more variation in the gait between steps.

This analysis covers one axis of the gyroscope. The visible differences can also be seen in the other two gyroscope axes and accelerometer signals. As the intact limb is closer to the non-amputee, it should be expected that it will perform more highly than the prosthetic side.

6.2 Baseline Model Performance

As before, a set of baselines are needed to determine if the personalisation methods result in a performance improvement. All baselines will be evaluated using the amputee test data sets. The two baselines will be the performance of the general models and the performance of models constructed from only the target amputee's data.

The general models were constructed in the previous Chapter from the large source data set of gait data, excluding subjects 1, 3 and 9. These are all 32 unit LSTM models. Both the trans-tibial amputee's right intact limb and left prosthetic limb were tested separately. The general models achieved a classification performance of $74.2\% \pm 9.4$ for the intact limb. For the amputated limb, the performance was significantly lower at $55.3\% \pm 9.6$. From the previous study non-amputee subjects achieved around 72%, this is comparably to the performance of the intact limb.

Table 6-2 shows the confusion matrix for the general model. The matrix shows that the

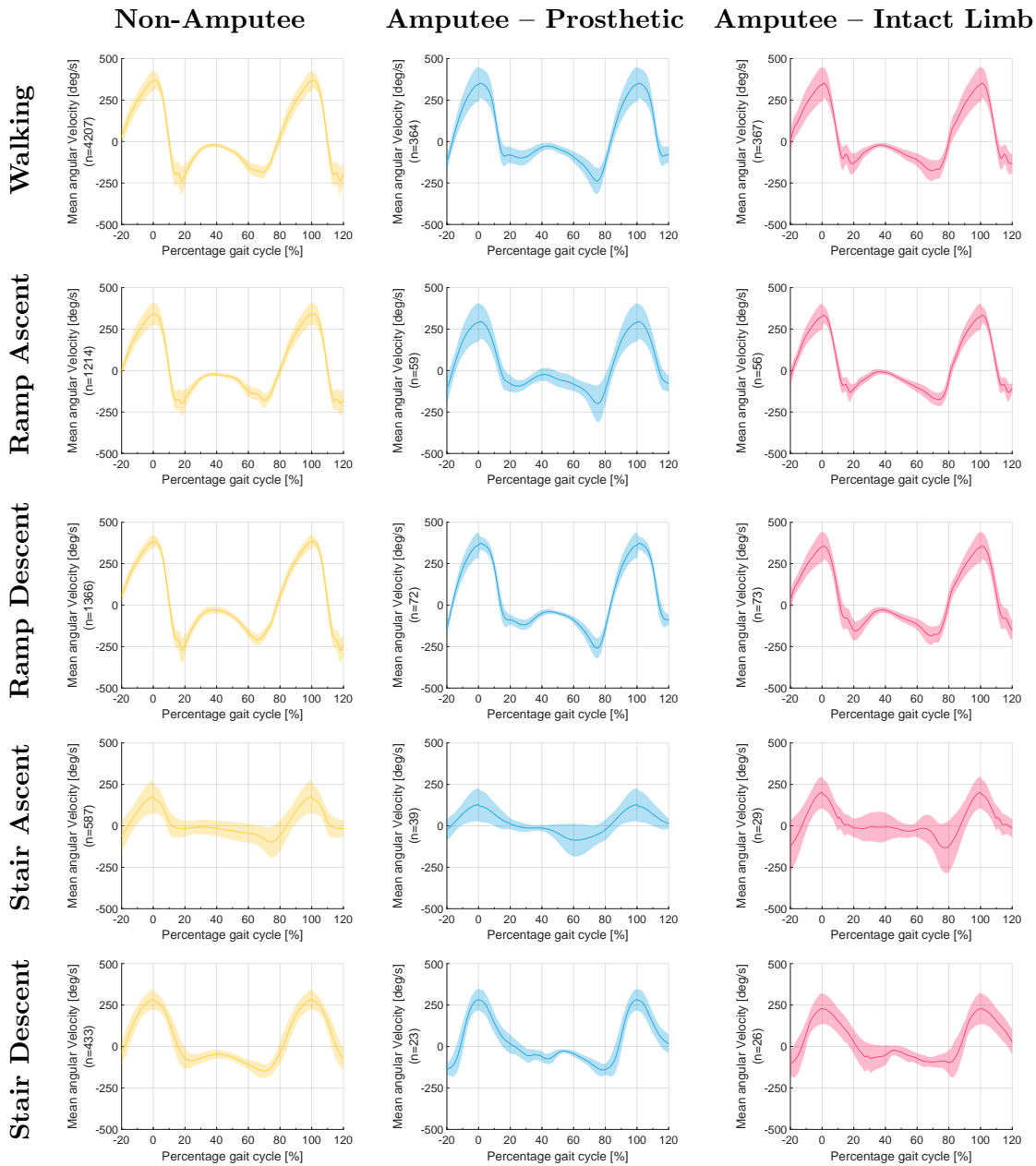


Figure 6-1: Angular velocity of the shank during different activities for an amputee and non-amputee. Data is for the sagittal Plane. The yellow line is for Non-Amputee (Subject 01 Left Ankle); The red line is the intact limb of the trans-tibial amputee; The blue line is the prosthetic of the trans-tibial amputee. The solid line shows the mean of the steps recorded for each activity. The filled area represents the standard deviation.

general mode performs poorly on walking and stair descent for both limbs. The performance for stair descent of the prosthetic limb is notable poor.

Table 6-2: Confusion matrix of general models presented with target subject test data. Columns represent the prediction labels, and the rows represent the actual labels. Each value represents the percentage of total predicted labels for that class. (Ramp Ascent (RA), Ramp Descent (RD), Stair Ascent (SA), Stair Descent (SD))

		(a) Prosthetic Limb					
		Predicted Classes					
		WALK	RA	RD	SA	SD	STOP
True Classes	WALK	46.6	12.8	9.0	1.7	23.1	5.9
	RA	8.5	77.7	0.0	9.8	13.4	10.6
	RD	26.0	6.4	91.0	0.3	33.7	2.1
	SA	0.0	0.4	0.0	88.1	0.0	1.0
	SD	14.5	2.6	0.0	0.1	29.3	18.0
	STOP	4.4	0.0	0.0	0.0	0.6	62.4

		(b) Intact Limb					
		Predicted Classes					
		WALK	RA	RD	SA	SD	STOP
True Classes	WALK	69.8	0.6	11.6	0.2	14.9	0.4
	RA	14.3	97.5	0.0	21.7	7.8	4.4
	RD	8.5	0.0	73.4	0.0	20.7	0.0
	SA	0.0	0.0	0.0	77.7	0.0	0.0
	SD	0.1	1.9	15.0	0.1	55.5	0.8
	STOP	7.3	0.0	0.0	0.2	1.1	94.4

The second baseline is a set of models trained using only the amputee data. Different quantities of training windows were used to provide performance metrics for various data amounts. Figure 6-2 shows the classification performance for both legs when tested with the test data sets. The full results of this experiment can be found in Appendix A Table A-7. The average number of epochs to train for all models was 7, with a 95th percentile of 9.

Figure 6-2 shows performance improving rapidly with increasing training windows levelling out after 500 samples. In all cases the prosthetic limb performs worse than the intact limb. With increasing windows the performance gap stays consistent.

Table 6-3 shows the confusion matrix for both limbs when a bespoke model was trained with 750 training windows. This is markedly better than the confusion matrix for the general model. This backs up the observations found in the literature that general models from non-amputees perform poorly for amputees[173, 175]. However, several classes across

both limbs perform worse than the general model, suggesting the general model contains knowledge that could be used to improve performance.

Table 6-3: confusion matrix for a bespoke amputee LMR model presented with test data. The 32 unit LSTM model was trained with 750 target data window. Columns represent the prediction labels and the rows represent the real labels. Each value represent the percentage of total predicted labels for that class. (Ramp Ascent (RA), Ramp Descent (RD), Stair Ascent (SA), Stair Descent (SD))

(a) Prosthetic Limb

		Predicted Classes					
		WALK	RA	RD	SA	SD	STOP
True Classes	WALK	68.2	0.6	42.6	0.6	0.0	0.0
	RA	9.7	94.0	3.8	1.4	0.5	0.0
	RD	18.9	0.4	53.6	0.2	0.0	0.0
	SA	0.0	0.0	0.0	79.4	0.0	0.0
	SD	0.0	0.0	0.0	0.0	99.5	0.0
	STOP	3.2	4.9	0.0	18.4	0.0	100.0

(b) Intact Limb

		Predicted Classes					
		WALK	RA	RD	SA	SD	STOP
True Classes	WALK	83.1	2.6	19.3	0.0	0.0	0.0
	RA	12.3	93.3	7.4	12.0	0.0	0.8
	RD	3.7	0.0	72.0	0.1	3.3	0.0
	SA	0.0	0.0	0.0	83.6	0.0	0.0
	SD	0.0	0.0	0.0	0.0	96.7	0.0
	STOP	0.9	4.1	1.2	4.3	0.0	99.2

6.3 Data supplementation

The first personalised model technique that will be investigated is data supplementation. This involves supplementing target data with a varying amount of data from a general source set to form a large training set. The source data is made up of a larger number of non-amputee subjects.

The experiment consisted of mixing 100, 250, 500 and 750 target data windows with between 100 and 3000 source windows. On average, each model took 10 epochs to train, 95th percentile of 17. In general, the more data used, the larger the number of epochs required. Table 6-4 shows classification performance for all combinations.

As before, the performance of the prosthetic side is lower than the intact side. All classification accuracies for the prosthetic exceed the general model performance. However, the

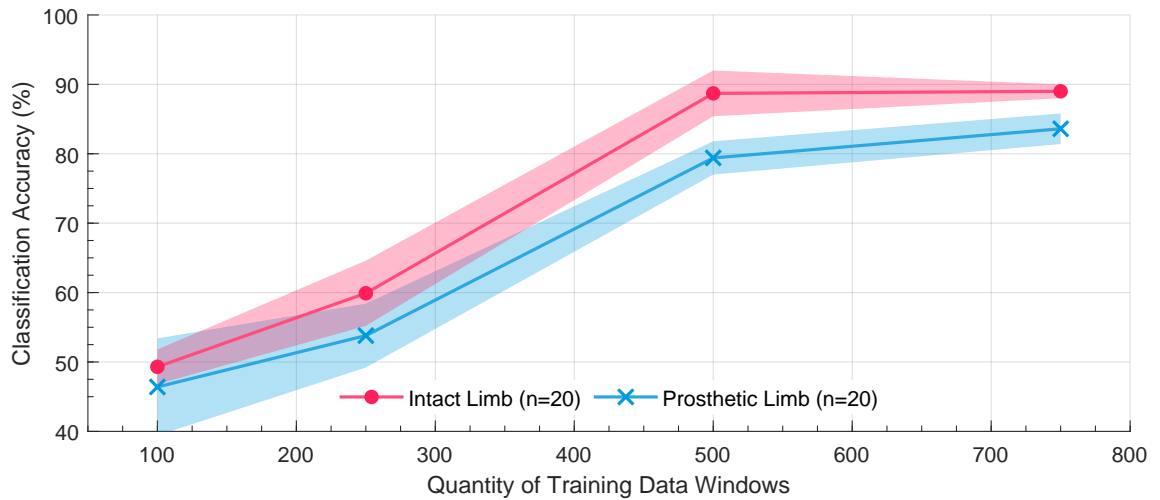


Figure 6-2: Classification accuracy for HAR model using increasing quantities of only amputee target data. The red line shows the performance of the trained model on the intact limb of a trans-tibial amputee. The blue line shows the performance of the trained model for the prosthetic side. The filled areas represent the standard deviation (n=20).

lowest two results for 100 source windows perform worse than the general model for the intact limb.

For the intact side, none of the 500 and 750 target window models exceeds the performance of the bespoke models with the same quantity of target windows. For the amputated side, all but 750 target window, 100 source window result exceed the bespoke model performance.

At a lower value of target windows, a higher quantity of source windows improves performance; less source data is needed at higher target windows. At higher values of target data windows, the performance improvement is minimal, especially for the intact side. This suggests that this method may become less valuable the more target data available.

6.4 Transfer Learning

Transfer learning involves using the knowledge captured in an existing model as a starting point to building a personalised model. For this experiment, five general models constructed in Chapter 5 were used as the starting point. Varying quantities of target amputee data windows were then used to fine-tune these starting models to personalise them to the amputee. By freezing the different network layers, attempts to reduce the computation load required to train the model could be made.

Figure 6-3 shows three different experiments in transfer learning. Each trained different

Table 6-4: Table of classification accuracy for amputee test data for a model trained using varying amounts of Source and Target training data. The cell value represents the percentage classification accuracy $\pm \sigma$ ($n = 8$). The highest classification accuracy for each quantity of target windows has been highlighted in bold

(a) Intact Limb

		Target Training Windows			
		100	250	500	750
Source Training Windows	0	0.428 \pm 0.05	0.448 \pm 0.05	0.863 \pm 0.02	0.869 \pm 0.01
	100	0.717 \pm 0.03	0.682 \pm 0.02	0.879 \pm 0.02	0.877 \pm 0.04
	250	0.764 \pm 0.04	0.730 \pm 0.03	0.883 \pm 0.03	0.889\pm0.01
	500	0.800 \pm 0.04	0.795 \pm 0.03	0.875 \pm 0.02	0.888 \pm 0.02
	750	0.815 \pm 0.01	0.801 \pm 0.02	0.873 \pm 0.02	0.881 \pm 0.02
	1000	0.801 \pm 0.04	0.786 \pm 0.03	0.886\pm0.03	0.874 \pm 0.02
	1500	0.835\pm0.03	0.794 \pm 0.06	0.871 \pm 0.01	0.875 \pm 0.03
	3000	0.825 \pm 0.01	0.826\pm0.07	0.846 \pm 0.03	0.863 \pm 0.03

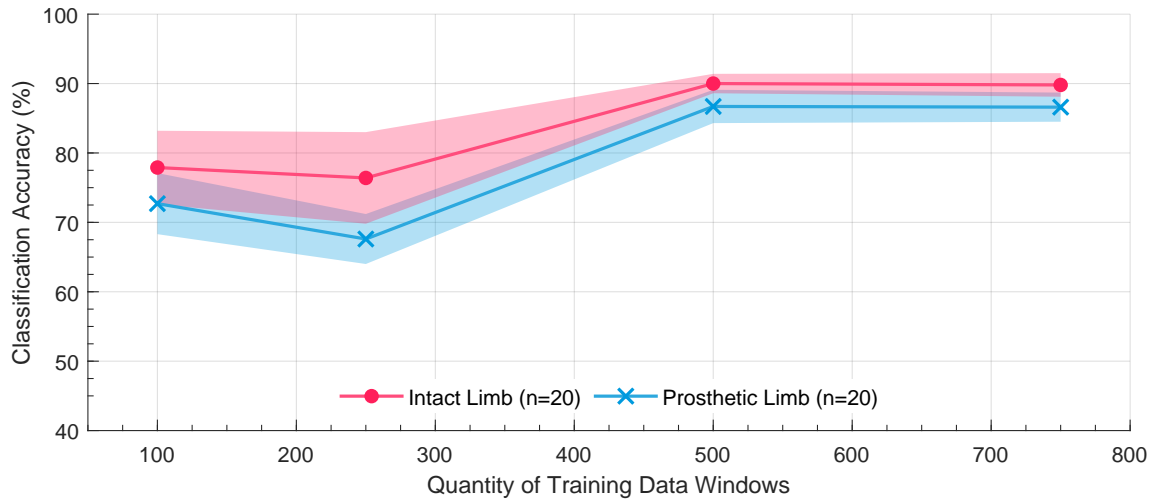
(b) Prosthetic Limb

		Target Training Windows			
		100	250	500	750
Source Training Windows	0	0.447 \pm 0.02	0.469 \pm 0.03	0.765 \pm 0.04	0.830 \pm 0.03
	100	0.626 \pm 0.05	0.643 \pm 0.03	0.855 \pm 0.02	0.813 \pm 0.04
	250	0.714 \pm 0.04	0.611 \pm 0.02	0.836 \pm 0.05	0.843 \pm 0.03
	500	0.752 \pm 0.03	0.729 \pm 0.08	0.842 \pm 0.02	0.840 \pm 0.05
	750	0.734 \pm 0.06	0.712 \pm 0.04	0.848 \pm 0.03	0.847 \pm 0.02
	1000	0.756 \pm 0.02	0.756 \pm 0.08	0.875\pm0.03	0.872\pm0.01
	1500	0.734 \pm 0.02	0.764\pm0.05	0.869 \pm 0.02	0.852 \pm 0.02
	3000	0.767\pm0.02	0.764\pm0.04	0.874 \pm 0.02	0.849 \pm 0.02

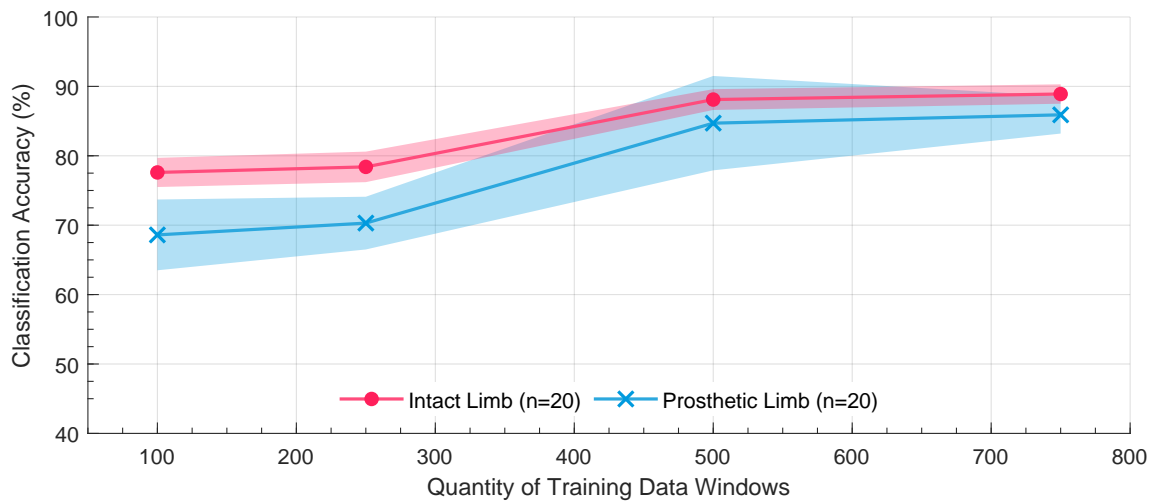
layers of the network. The first trained all layers, the second just the dense layer, and finally just the LSTM layer.

On average, each model took 6 epochs 95th percentile of 10. In general, the more data used, the larger the number of epochs required.

When training all layers, classification performance significantly increased over the base general model with only a few target windows. For the prosthetic side with 100 target windows, there was a 22% increase in performance over the general model. This reduced to just under 1% for the intact limb at 750 windows. The improvement over the bespoke model slightly higher achieving at least a 3% improvement. Overall transfer learning resulted in an improvement for all configurations.

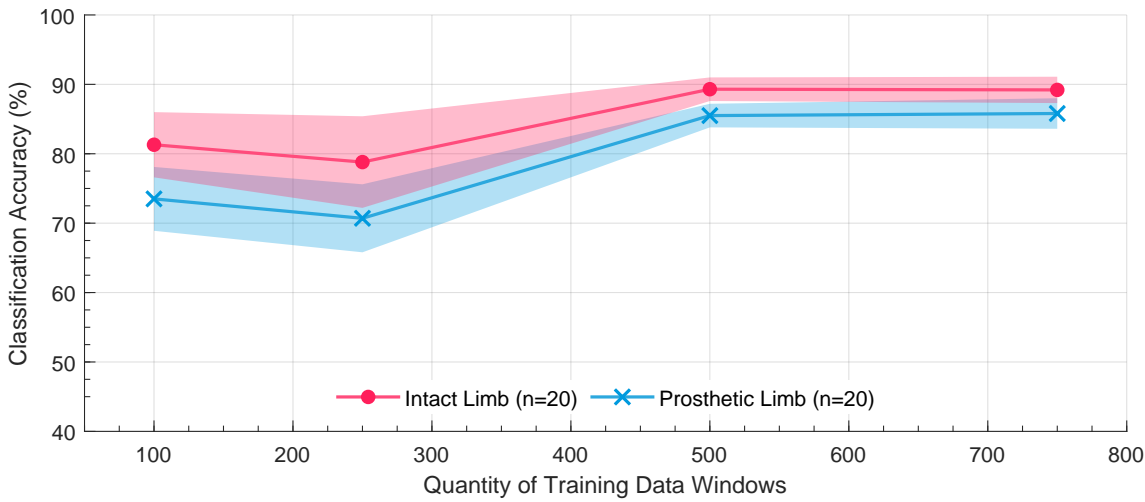


(a) Fine tuning all layers



(b) Fine tuning only the dense layer

Figure 6-3: Classification accuracy for re-training a pre-trained model using increasing quantities of amputee target data. The red line shows the performance of the trained model on the intact limb of a trans-tibial amputee. The blue line shows the performance of the trained model for the prosthetic side. The filled areas represent the standard deviation (n=20).



(c) Fine tuning only the LSTM layer

Figure 6-3: Classification accuracy for re-training a pre-trained model using increasing quantities of amputee target data (Cont.).

Fine-tuning only the dense layer did not result in better performance than fine-tuning all layers and for the intact limb at 500 and 750 windows was worse than the baseline bespoke model. Using this method significantly increased standard deviation for the prosthetic limb, although slightly reduced σ for the intact limb.

Fine-tuning only the LSTM layer gave the best performance for 100 and 250 target windows compared to fine-tuning all layers. Performance was a couple of percent better; however, the higher two target window quantities performance was approximately a percent worse. The standard deviation remained roughly the same. On balance, it showed no improvement over fine-tuning all layers.

6.5 Discussion and Conclusions

The work in this Chapter set out to investigate if the personalisation methods for an LMR classifier developed in Chapter 5 were applicable to an amputee. A set of comparable data was collected from a trans-tibial amputee to achieve this. This was then used to repeat the previous experiments for the amputee subject.

Jamieson and Lonini suggested that the direct use of a general model trained using only non-amputee data would not perform adequately for a person with gait impairments[173, 175]. This was borne out in the results when data from the prosthetic limb was tested. It achieved a classification accuracy of 55.3%, significantly less than the non-amputee subjects.

When data from the amputee’s intact limb was used, classification performance was much higher. The test subject has a significant asymmetry in their gait, with the intact side matching more closely to the source non-amputees gait. This is an area that requires additional research as it is potentially an easy way to improve the performance of an amputee gait classifier.

Both the data supplementation and transfer learning approaches improved over the baseline classifiers. The differences observed in the previous Chapter were again shown. The quantity of source data required for data supplementation was hard to predict, and only training specific layers for transfer learning resulted in minimal changes. As before, the transfer learning performance appeared to perform more consistently and required significantly less computing resources to train.

The overall performance of both the bespoke model and personalised models at 500 and 750 training windows was significantly higher than seen in the previous Chapter. They also looked more likely to be levelling off in performance. There are two likely reasons for this. First, a much smaller testing set was used, so the trained model was tested with a narrower range of environments. Secondly, all the amputee data was collected at a single site, and therefore the training data is likely to include environments seen in the test set. The division of stairs episodes would also have a similar effect.

Further, as data for only one amputee was collected, it is not possible to say whether these methods are more generally applicable. To test the results further, more amputees and more data per amputee are required.

In the previous Chapter, a method for improving data supplementation was data grouping. It is unlikely to be successful in this scenario due to the likely difficulty in finding subjects with a similar gait to an amputee. However, there may be potential in including more persons with gait impairments to improve the networks ability to adapt. This is a potential area for further research.

No literature was found investigating the classification performance difference between the two lower limbs of an amputee or other person with asymmetric gait. The results consistently showed that classification of the intact limb had higher accuracies than the amputated limb. This could be an exciting area of research for improving the classifier’s performance using both sides of the body. A form of ensemble network could be a good candidate for this.

The personalisation approach shows promise in the area of amputee gait classification. As concluded in the previous chapter the general training population should ideally contain

individuals of similar gait to the target subject. This is significantly harder for amputees. Further research is required in this area to exploit the potential for using non-amputee data to reduce data requirements for amputees and to investigate the limitations of this approach. Further research into the use of the intact limb to improve classification performance is also required,

Chapter 7

Conclusions and Further Work

In this Chapter, the work conducted towards developing an ML method for an activity classifier for an amputee to improve locomotion mode selection in the powered prosthesis is briefly discussed. This is followed by some suggestions for further work to continue this research.

7.1 Conclusions

An effort was made to contribute to the development of a locomotion mode classifier for amputees in the hope that this will be useful to the improvement of powered prosthetic devices. Specifically, the hypothesis stated in the Introductory Chapter was:

A Machine Learning approach based on Long Short Term Memory (LSTM) architecture can be used to predict gait modes with data requirements reduced through a transfer learning approach.

A literature survey was conducted to establish the background around gait, lower limb powered prosthesis, and Machine Learning methods. The need for a system to identify an amputee's current locomotive intent to inform the selection of appropriate control modes was established. The current state of the art research has not demonstrated IMU-based ML classifiers applied to amputees; however, they have been applied to individuals without gait impairment. This appears to be partly due to the difficulties in collecting a large enough data set of amputee gait data. But also due to the highly individual nature of gait, especially for amputees. The need for further research in this area was established, especially for ways to reduce data requirements and increase performance.

In order to address these research gaps, a large set of gait data is required from non-amputees and amputees. As IMU gait data for lower limb amputees could not be found, a new data set is required. A novel system comprised of wireless IMUs and a companion smartphone app was developed to achieve this. This allowed for the unsupervised collection of labelled natural gait data from numerous individuals. Methods were developed for post-processing the gait data to prepare it for use in a TensorFlow ML environment. Work was also undertaken to develop the TensorFlow ML environment.

A journal article published in *Sensors* was presented that investigated the internal behaviour of an LSTM based LMR classifier. A public dataset of 22 individuals was collected using the previous Chapter's methodology. This data set was used to analyse the internal behaviour of a reduced complexity LSTM network. Experiments around analysing the network's hidden state were undertaken to establish a link between the input data and output classification. The analysis identifies that the model primarily classifies activity type based on data around early stance. Additional work was undertaken on a full LSTM LMR network to identify activity types for unseen novel subjects. Evaluation of generalisation for unseen subjects reveals low sensitivity to hyper-parameters with issues caused by over-fitting individuals' gait traits. Although an accuracy of greater than 95% is possible for a seen individual classification, the network struggled to classify unseen individuals, achieving around 80%. Investigating the differences between individual subjects showed that gait variations between users primarily occur in early stance, potentially explaining the poor generalisation. Adjustment of hyper-parameters alone could not solve this, demonstrating the need for individual personalisation of models.

Based on the need for individual personalisation, methods for achieving this were investigated. A survey of literature revealed that transfer learning is a promising approach. However, its applicability to real-world data has not been investigated, nor has the requirements for the quantity of a target individual's data. Additional data for three subjects was collected using the previous methods developed. This allowed for an extensive study of the benefits of transfer learning with different quantities of target subject data. In order to use the unstructured real-world continuous data, new methods for data division were required. The data was poorly distributed, and therefore data rebalancing was required. This was accomplished by dividing the data into episodes, each containing a single continuous period of one activity. By combining episodes, a balanced data set could be constructed. This also had the benefit of allowing for multiple data sets to be systematically built and ensuring that the test data set was unseen.

A set of baselines were developed to compare the network's performance against. These were the best performance that could be achieved by either a general model built for other's

data or a fully bespoke model constructed for a target subject's data. Two personalisation methods were attempted, data supplementation and transfer learning. Both methods improved the performance of the target subject's classification, achieving 90% maximum. The maximum improvements were seen at low quantities of target data, demonstrating both methods showed promise for situations where only limited data is available.

Following the development of the personalisation methods, their applicability to amputees was evaluated. The literature found that studies had struggled to classify amputees using gait data from non-amputees. However, no study had attempted personalisation techniques using amputee data. There was also no investigation into differences in how the amputated and intact limbs would perform. Amputees have a very asymmetric gait; therefore, it should be expected that both legs would have different classification performances. A small set of amputee data was collected from a single trans-tibial amputee to investigate personalisation methods. Using the methodology previously discussed, personalisation experiments were repeated. The results showed a dramatic improvement in classification performance using limited amputee data. Both personalisation methods worked, achieving 90% accuracy for the intact limb and just under 90% for the amputated limb. Due to the limited data available for testing, it is not possible to say how generally these methods work, but they appear promising and should be investigated further.

This work has raised many additional research questions. Such as the applicability of the amputee personalisation result to people with other gait abnormalities. There is future research in to the use of an amputees intact limb classifier to improve gait classification. These form areas that could be focus of future research.

The work allows for large scale collection of gait data that is free from laboratory constraints while reducing gait bias from researchers. The work also highlighted the improvements in classification performance that can be achieved by using a general population as a starting point for building a bespoke classifier. Initial work indicates that this is highly applicable to classifying individual with gait abnormalities. Further work is still required to see how generally applicable this is to amputees and other gait abnormalities.

7.2 Further Work

The work conducted has demonstrated that this is an area of research with promise. However, there are numerous aspects where additional research could yield improvements. These include:

- Additional amputees – The trial only included a single trans-tibial amputee. Addi-

tional amputee trials should involve multiple subjects, including amputees of different weights, heights, and levels of amputation in multiple environments to test the applicability more generally.

- Investigate how the transition will affect performance – Due to the data division scheme employed for the personality study, the transition between activities was not considered. In Chapter 4 transition was identified as a key area of error. Looking at how classification performance changes around this area would inform knowledge of the current performance of the model during transition. Changes to the labelling of transitions, ML model architecture, and model hyper-parameters should then be investigated to see if transition performance can be improved.
- A greater number of environments – It was hypothesised that the addition of new environments affects performance; however, due to the limited data labelling, it was not possible to investigate this. Modifying the app to store location data would allow for greater detail to be understood about the environment. By using map data the type of environment could be automatically tagged. Classification performance could then be filtered to see how performance changes based on different environments.
- Investigate more complex LMR networks – A very shallow LSTM network was investigated. This was selected due to its adequate performance and ability to iterate quickly with a shallow network. Other work has successfully used deeper and more sophisticated networks; this should be explored further. There is also a large scope for hyper-parameter optimisation as only a little work was performed in this area.
- Implementation for real-time – The fact that the system works on a smartphone means that a system could be deployed in the real world for more extensive testing. Direct real-time feedback on the models performance could be used to inform a semi-supervised learning system. This would allow for far greater data quantities to be used during training without the need for direct user labelling.

Appendix A

Tables of Results

This Appendix contains tables of data for Chapter 5 which investigated personalisation methods for LSTM models using data for non-amputees.

A.1 Tables of Model Performance for Non-Amputee Bespoke Target Model

The following section contains tables of data for investigation into classification performance for LSTM models trained using increasing amounts of a target subject data. This section contains Tables A-1, A-2 and A-3

Table A-1: Table of results for classification performance of different size LSTM networks trained with varying amount of target subject data for subject 1. The table shows the classification accuracy for the target user training, validation and test data sets $\pm\sigma(n = 25)$. A value of one represents 100% correct classification.

		6 Units				16 Units			
		Train	Valid	Test	Epochs	Train	Valid	Test	Epochs
Target Training Windows	100	0.697 \pm 0.10	0.477 \pm 0.10	0.492 \pm 0.07	36	0.776 \pm 0.04	0.477 \pm 0.04	0.527 \pm 0.03	22
	250	0.765 \pm 0.08	0.543 \pm 0.09	0.570 \pm 0.10	22	0.805 \pm 0.09	0.542 \pm 0.07	0.607 \pm 0.09	12
	500	0.783 \pm 0.09	0.599 \pm 0.05	0.655 \pm 0.09	12	0.875 \pm 0.03	0.641 \pm 0.03	0.743 \pm 0.06	9
	750	0.835 \pm 0.04	0.623 \pm 0.06	0.682 \pm 0.06	10	0.899 \pm 0.05	0.617 \pm 0.03	0.726 \pm 0.05	7
	1000	0.821 \pm 0.04	0.713 \pm 0.04	0.736 \pm 0.05	15	0.854 \pm 0.02	0.704 \pm 0.02	0.782 \pm 0.04	8
	1500	0.816 \pm 0.02	0.713 \pm 0.02	0.752 \pm 0.05	17	0.864 \pm 0.02	0.689 \pm 0.02	0.752 \pm 0.04	8
	3000	0.823 \pm 0.03	0.777 \pm 0.04	0.753 \pm 0.06	18	0.864 \pm 0.03	0.763 \pm 0.03	0.752 \pm 0.05	11
	6000	0.840 \pm 0.03	0.771 \pm 0.06	0.750 \pm 0.06	24	0.909 \pm 0.03	0.776 \pm 0.04	0.789 \pm 0.05	12
	9000	0.848 \pm 0.02	0.718 \pm 0.03	0.741 \pm 0.04	18	0.903 \pm 0.02	0.732 \pm 0.03	0.788 \pm 0.05	11
	12000	0.851 \pm 0.03	0.738 \pm 0.05	0.784 \pm 0.05	23	0.907 \pm 0.02	0.742 \pm 0.03	0.814 \pm 0.04	13
	15000	0.813 \pm 0.04	0.732 \pm 0.06	0.783 \pm 0.04	23	0.889 \pm 0.02	0.764 \pm 0.03	0.844 \pm 0.03	15

		32 Units				64 Units			
		Train	Valid	Test	Epochs	Train	Valid	Test	Epochs
Target Training Windows	100	0.801 \pm 0.05	0.497 \pm 0.05	0.566 \pm 0.04	16	0.724 \pm 0.09	0.439 \pm 0.07	0.521 \pm 0.08	9
	250	0.815 \pm 0.08	0.531 \pm 0.03	0.622 \pm 0.05	8	0.869 \pm 0.04	0.540 \pm 0.03	0.650 \pm 0.03	7
	500	0.883 \pm 0.03	0.631 \pm 0.03	0.720 \pm 0.05	6	0.907 \pm 0.03	0.615 \pm 0.04	0.746 \pm 0.03	5
	750	0.934 \pm 0.02	0.633 \pm 0.02	0.749 \pm 0.03	5	0.932 \pm 0.02	0.627 \pm 0.02	0.727 \pm 0.04	4
	1000	0.888 \pm 0.02	0.696 \pm 0.02	0.774 \pm 0.04	7	0.886 \pm 0.02	0.694 \pm 0.02	0.765 \pm 0.03	5
	1500	0.865 \pm 0.01	0.704 \pm 0.03	0.758 \pm 0.04	6	0.885 \pm 0.02	0.693 \pm 0.02	0.771 \pm 0.02	5
	3000	0.895 \pm 0.04	0.752 \pm 0.02	0.767 \pm 0.02	8	0.921 \pm 0.03	0.754 \pm 0.02	0.771 \pm 0.01	8
	6000	0.924 \pm 0.03	0.758 \pm 0.06	0.781 \pm 0.06	10	0.937 \pm 0.01	0.719 \pm 0.03	0.776 \pm 0.03	8
	9000	0.923 \pm 0.03	0.710 \pm 0.05	0.778 \pm 0.05	9	0.917 \pm 0.02	0.709 \pm 0.03	0.752 \pm 0.04	7
	12000	0.920 \pm 0.04	0.712 \pm 0.03	0.798 \pm 0.01	10	0.902 \pm 0.04	0.712 \pm 0.04	0.779 \pm 0.03	7
	15000	0.903 \pm 0.02	0.780 \pm 0.04	0.839 \pm 0.03	17	0.889 \pm 0.06	0.743 \pm 0.06	0.805 \pm 0.06	10

Table A-2: Table of results for classification performance of different size LSTM networks trained with varying amount of target subject data for subject 3. The table shows the classification accuracy for the target user training, validation and test data sets $\pm\sigma(n = 25)$. A value of one represents 100% correct classification.

		6 Units				16 Units			
		Train	Valid	Test	Epochs	Train	Valid	Test	Epochs
Target Training Windows	100	0.979 \pm 0.03	0.765 \pm 0.04	0.628 \pm 0.06	49	0.986 \pm 0.02	0.790 \pm 0.05	0.612 \pm 0.05	29
	250	0.968 \pm 0.04	0.772 \pm 0.08	0.607 \pm 0.06	22	0.997 \pm 0.01	0.809 \pm 0.04	0.638 \pm 0.05	16
	500	0.993 \pm 0.01	0.891 \pm 0.03	0.684 \pm 0.04	19	0.999 \pm 0.00	0.898 \pm 0.02	0.701 \pm 0.04	12
	750	0.997 \pm 0.00	0.875 \pm 0.04	0.680 \pm 0.04	14	0.999 \pm 0.00	0.905 \pm 0.02	0.692 \pm 0.04	8
	1000	0.986 \pm 0.01	0.900 \pm 0.04	0.668 \pm 0.03	25	0.998 \pm 0.00	0.929 \pm 0.02	0.694 \pm 0.05	13
	1500	0.967 \pm 0.02	0.828 \pm 0.03	0.687 \pm 0.05	17	0.997 \pm 0.00	0.873 \pm 0.04	0.682 \pm 0.04	15
	3000	0.762 \pm 0.09	0.420 \pm 0.06	0.589 \pm 0.07	7	0.865 \pm 0.05	0.460 \pm 0.05	0.648 \pm 0.04	5
	6000	0.840 \pm 0.04	0.639 \pm 0.02	0.701 \pm 0.04	14	0.870 \pm 0.04	0.626 \pm 0.05	0.710 \pm 0.03	8
	9000	0.829 \pm 0.05	0.647 \pm 0.03	0.724 \pm 0.03	18	0.904 \pm 0.03	0.697 \pm 0.03	0.770 \pm 0.02	10
	12000	0.835 \pm 0.03	0.802 \pm 0.04	0.802 \pm 0.02	31	0.917 \pm 0.02	0.851 \pm 0.02	0.853 \pm 0.02	17
	15000	0.819 \pm 0.02	0.717 \pm 0.03	0.811 \pm 0.01	40	0.902 \pm 0.02	0.768 \pm 0.03	0.856 \pm 0.02	15

		32 Units				64 Units			
		Train	Valid	Test	Epochs	Train	Valid	Test	Epochs
Target Training Windows	100	0.986 \pm 0.02	0.777 \pm 0.05	0.634 \pm 0.07	20	0.970 \pm 0.03	0.763 \pm 0.05	0.624 \pm 0.05	13
	250	0.997 \pm 0.01	0.809 \pm 0.03	0.629 \pm 0.03	10	0.993 \pm 0.01	0.814 \pm 0.03	0.653 \pm 0.02	7
	500	1.000 \pm 0.00	0.918 \pm 0.01	0.715 \pm 0.03	9	1.000 \pm 0.00	0.916 \pm 0.02	0.727 \pm 0.03	7
	750	1.000 \pm 0.00	0.908 \pm 0.02	0.713 \pm 0.03	7	1.000 \pm 0.00	0.903 \pm 0.01	0.732 \pm 0.03	6
	1000	1.000 \pm 0.00	0.924 \pm 0.02	0.707 \pm 0.03	10	1.000 \pm 0.00	0.924 \pm 0.01	0.721 \pm 0.03	8
	1500	0.999 \pm 0.00	0.895 \pm 0.02	0.724 \pm 0.03	12	0.999 \pm 0.00	0.877 \pm 0.03	0.713 \pm 0.02	8
	3000	0.932 \pm 0.02	0.450 \pm 0.06	0.663 \pm 0.04	5	0.949 \pm 0.01	0.432 \pm 0.04	0.664 \pm 0.03	4
	6000	0.912 \pm 0.02	0.645 \pm 0.02	0.740 \pm 0.02	6	0.931 \pm 0.02	0.644 \pm 0.04	0.749 \pm 0.02	5
	9000	0.934 \pm 0.01	0.714 \pm 0.02	0.790 \pm 0.01	8	0.940 \pm 0.02	0.718 \pm 0.03	0.789 \pm 0.02	7
	12000	0.934 \pm 0.02	0.869 \pm 0.01	0.874 \pm 0.01	14	0.948 \pm 0.02	0.862 \pm 0.03	0.869 \pm 0.02	13
	15000	0.922 \pm 0.02	0.792 \pm 0.04	0.881 \pm 0.02	13	0.934 \pm 0.02	0.798 \pm 0.03	0.885 \pm 0.02	11

Table A-3: Table of results for classification performance of different size LSTM networks trained with varying amount of target subject data for subject 9. The table shows the classification accuracy for the target user training, validation and test data sets $\pm\sigma(n = 25)$. A value of one represents 100% correct classification.

		6 Units				16 Units			
		Train	Valid	Test	Epochs	Train	Valid	Test	Epochs
Target Training Windows	100	0.859 \pm 0.13	0.648 \pm 0.10	0.574 \pm 0.08	62	0.929 \pm 0.04	0.704 \pm 0.04	0.657 \pm 0.03	30
	250	0.896 \pm 0.03	0.631 \pm 0.04	0.666 \pm 0.06	28	0.935 \pm 0.02	0.671 \pm 0.05	0.687 \pm 0.03	16
	500	0.920 \pm 0.05	0.649 \pm 0.06	0.679 \pm 0.04	19	0.953 \pm 0.03	0.676 \pm 0.05	0.724 \pm 0.03	10
	750	0.921 \pm 0.03	0.676 \pm 0.06	0.688 \pm 0.04	13	0.977 \pm 0.01	0.742 \pm 0.05	0.728 \pm 0.07	8
	1000	0.880 \pm 0.04	0.677 \pm 0.07	0.685 \pm 0.04	21	0.946 \pm 0.04	0.732 \pm 0.06	0.708 \pm 0.05	10
	1500	0.907 \pm 0.03	0.747 \pm 0.05	0.718 \pm 0.04	21	0.962 \pm 0.02	0.793 \pm 0.03	0.750 \pm 0.03	11
	3000	0.884 \pm 0.01	0.845 \pm 0.05	0.698 \pm 0.05	53	0.957 \pm 0.02	0.914 \pm 0.05	0.725 \pm 0.03	24
	6000	0.897 \pm 0.03	0.752 \pm 0.05	0.739 \pm 0.04	20	0.956 \pm 0.01	0.783 \pm 0.04	0.764 \pm 0.02	14
	9000	0.789 \pm 0.06	0.602 \pm 0.05	0.753 \pm 0.02	11	0.902 \pm 0.04	0.683 \pm 0.04	0.821 \pm 0.03	9
	12000	0.825 \pm 0.06	0.672 \pm 0.05	0.751 \pm 0.03	18	0.909 \pm 0.03	0.702 \pm 0.02	0.819 \pm 0.03	9
	15000	0.840 \pm 0.02	0.726 \pm 0.02	0.731 \pm 0.04	18	0.912 \pm 0.02	0.772 \pm 0.03	0.759 \pm 0.02	11

		32 Units				64 Units			
		Train	Valid	Test	Epochs	Train	Valid	Test	Epochs
Target Training Windows	100	0.924 \pm 0.03	0.677 \pm 0.03	0.658 \pm 0.04	19	0.894 \pm 0.03	0.644 \pm 0.05	0.627 \pm 0.03	12
	250	0.943 \pm 0.04	0.646 \pm 0.06	0.702 \pm 0.04	11	0.926 \pm 0.02	0.607 \pm 0.04	0.690 \pm 0.01	7
	500	0.975 \pm 0.02	0.706 \pm 0.06	0.745 \pm 0.03	8	0.977 \pm 0.02	0.694 \pm 0.03	0.722 \pm 0.02	7
	750	0.979 \pm 0.02	0.745 \pm 0.04	0.741 \pm 0.04	6	0.991 \pm 0.01	0.747 \pm 0.03	0.756 \pm 0.03	6
	1000	0.977 \pm 0.01	0.767 \pm 0.02	0.760 \pm 0.04	8	0.977 \pm 0.02	0.769 \pm 0.04	0.745 \pm 0.06	8
	1500	0.972 \pm 0.01	0.823 \pm 0.02	0.782 \pm 0.03	8	0.968 \pm 0.02	0.814 \pm 0.01	0.771 \pm 0.03	7
	3000	0.978 \pm 0.00	0.945 \pm 0.01	0.759 \pm 0.03	17	0.978 \pm 0.01	0.949 \pm 0.01	0.754 \pm 0.02	15
	6000	0.963 \pm 0.01	0.794 \pm 0.02	0.781 \pm 0.03	10	0.966 \pm 0.00	0.768 \pm 0.03	0.762 \pm 0.03	8
	9000	0.927 \pm 0.02	0.686 \pm 0.02	0.825 \pm 0.03	7	0.941 \pm 0.02	0.699 \pm 0.02	0.841 \pm 0.02	6
	12000	0.933 \pm 0.02	0.710 \pm 0.01	0.826 \pm 0.03	7	0.939 \pm 0.02	0.719 \pm 0.01	0.816 \pm 0.03	6
	15000	0.921 \pm 0.04	0.775 \pm 0.03	0.756 \pm 0.05	10	0.956 \pm 0.01	0.801 \pm 0.03	0.785 \pm 0.04	9

A.2 Tables of Model Performance for Non-Amputee Transfer Learning

This section contains data tables for an investigation into model performance improvements by additional training of a general LSTM HAR model with varying amounts of target subject data. Tables A-4, A-5 and A-6 are included in this section.

Table A-4: Classification accuracy for a 32 unit LSTM model retrained with increasing amount of target data. The table shows the classification accuracy for the target user training, validation and test data sets $\pm\sigma(n = 25)$. A value of one represents 100% correct classification.

(a) Subject 01

Target Windows	Training	Validation	Testing	Average Epochs
100	0.962 \pm 0.02	0.692 \pm 0.05	0.781 \pm 0.03	4
250	0.953 \pm 0.02	0.658 \pm 0.04	0.791 \pm 0.03	4
500	0.963 \pm 0.01	0.715 \pm 0.03	0.805 \pm 0.03	5
750	0.950 \pm 0.01	0.732 \pm 0.03	0.812 \pm 0.03	4
1000	0.958 \pm 0.01	0.770 \pm 0.03	0.838 \pm 0.02	4
1500	0.950 \pm 0.02	0.773 \pm 0.04	0.837 \pm 0.02	4
3000	0.959 \pm 0.01	0.818 \pm 0.01	0.829 \pm 0.01	5
6000	0.957 \pm 0.01	0.793 \pm 0.04	0.832 \pm 0.02	5
9000	0.960 \pm 0.01	0.754 \pm 0.04	0.831 \pm 0.02	5
12000	0.959 \pm 0.01	0.781 \pm 0.02	0.831 \pm 0.02	5
15000	0.931 \pm 0.01	0.803 \pm 0.03	0.856 \pm 0.02	5

(b) Subject 03

Target Windows	Training	Validation	Testing	Average Epochs
100	1.000 \pm 0.00	0.956 \pm 0.02	0.803 \pm 0.03	6
250	1.000 \pm 0.00	0.953 \pm 0.02	0.803 \pm 0.02	6
500	1.000 \pm 0.00	0.969 \pm 0.01	0.827 \pm 0.03	6
750	1.000 \pm 0.00	0.970 \pm 0.01	0.824 \pm 0.02	6
1000	1.000 \pm 0.00	0.982 \pm 0.01	0.824 \pm 0.03	6
1500	1.000 \pm 0.00	0.976 \pm 0.01	0.825 \pm 0.03	7
3000	0.990 \pm 0.00	0.591 \pm 0.06	0.772 \pm 0.04	4
6000	0.982 \pm 0.00	0.729 \pm 0.02	0.855 \pm 0.02	4
9000	0.970 \pm 0.00	0.830 \pm 0.02	0.852 \pm 0.01	4
12000	0.965 \pm 0.01	0.892 \pm 0.02	0.905 \pm 0.01	8
15000	0.950 \pm 0.01	0.828 \pm 0.02	0.901 \pm 0.01	5

(c) Subject 09

Target Windows	Training	Validation	Testing	Average Epochs
100	0.998 \pm 0.00	0.851 \pm 0.05	0.802 \pm 0.03	4
250	0.994 \pm 0.00	0.841 \pm 0.02	0.840 \pm 0.02	4
500	0.992 \pm 0.00	0.848 \pm 0.03	0.847 \pm 0.02	4
750	0.987 \pm 0.00	0.869 \pm 0.02	0.849 \pm 0.02	4
1000	0.986 \pm 0.00	0.868 \pm 0.02	0.851 \pm 0.02	4
1500	0.985 \pm 0.00	0.872 \pm 0.02	0.857 \pm 0.01	4
3000	0.981 \pm 0.01	0.956 \pm 0.02	0.761 \pm 0.04	9
6000	0.973 \pm 0.00	0.807 \pm 0.03	0.787 \pm 0.04	4
9000	0.969 \pm 0.00	0.720 \pm 0.02	0.847 \pm 0.03	4
12000	0.965 \pm 0.00	0.734 \pm 0.02	0.847 \pm 0.03	4
15000	0.960 \pm 0.00	0.815 \pm 0.01	0.824 \pm 0.02	4

Table A-5: Classification accuracy for a 32 unit LSTM model with the dense layer retrained with increasing amount of target data. The table shows the classification accuracy for the target user training, validation and test data sets $\pm\sigma(n = 25)$. A value of one represents 100% correct classification.

(a) Subject 01

Target Windows	Training	Validation	Testing	Average Epochs
100	0.885 \pm 0.02	0.741 \pm 0.04	0.777 \pm 0.03	5
250	0.866 \pm 0.03	0.714 \pm 0.04	0.783 \pm 0.04	4
500	0.887 \pm 0.03	0.722 \pm 0.03	0.774 \pm 0.04	4
750	0.879 \pm 0.03	0.720 \pm 0.03	0.779 \pm 0.03	4
1000	0.878 \pm 0.03	0.758 \pm 0.01	0.806 \pm 0.03	5
1500	0.871 \pm 0.03	0.762 \pm 0.02	0.803 \pm 0.03	4
3000	0.857 \pm 0.02	0.843 \pm 0.02	0.821 \pm 0.02	5
6000	0.898 \pm 0.01	0.836 \pm 0.02	0.808 \pm 0.03	6
9000	0.926 \pm 0.01	0.754 \pm 0.03	0.821 \pm 0.02	12
12000	0.912 \pm 0.01	0.765 \pm 0.03	0.829 \pm 0.02	9
15000	0.892 \pm 0.01	0.779 \pm 0.02	0.840 \pm 0.01	9

(b) Subject 03

Target Windows	Training	Validation	Testing	Average Epochs
100	1.000 \pm 0.00	0.915 \pm 0.03	0.818 \pm 0.03	7
250	0.999 \pm 0.00	0.918 \pm 0.03	0.824 \pm 0.03	6
500	0.998 \pm 0.00	0.945 \pm 0.03	0.848 \pm 0.03	12
750	0.994 \pm 0.01	0.946 \pm 0.03	0.851 \pm 0.03	11
1000	1.000 \pm 0.00	0.982 \pm 0.01	0.837 \pm 0.04	26
1500	1.000 \pm 0.00	0.979 \pm 0.01	0.839 \pm 0.03	27
3000	0.962 \pm 0.01	0.729 \pm 0.07	0.811 \pm 0.02	4
6000	0.951 \pm 0.01	0.765 \pm 0.01	0.870 \pm 0.02	5
9000	0.943 \pm 0.01	0.848 \pm 0.01	0.870 \pm 0.01	6
12000	0.926 \pm 0.01	0.864 \pm 0.01	0.897 \pm 0.02	12
15000	0.918 \pm 0.01	0.838 \pm 0.03	0.890 \pm 0.01	9

(c) Subject 09

Target Windows	Training	Validation	Testing	Average Epochs
100	0.984 \pm 0.01	0.864 \pm 0.02	0.815 \pm 0.02	7
250	0.970 \pm 0.01	0.818 \pm 0.03	0.832 \pm 0.02	6
500	0.962 \pm 0.01	0.822 \pm 0.04	0.838 \pm 0.01	6
750	0.958 \pm 0.01	0.857 \pm 0.02	0.840 \pm 0.01	7
1000	0.957 \pm 0.01	0.867 \pm 0.02	0.846 \pm 0.02	7
1500	0.953 \pm 0.01	0.869 \pm 0.02	0.845 \pm 0.02	8
3000	0.938 \pm 0.01	0.914 \pm 0.01	0.783 \pm 0.04	10
6000	0.946 \pm 0.01	0.802 \pm 0.03	0.807 \pm 0.04	8
9000	0.932 \pm 0.01	0.734 \pm 0.03	0.856 \pm 0.02	6
12000	0.933 \pm 0.01	0.744 \pm 0.03	0.859 \pm 0.02	6
15000	0.920 \pm 0.01	0.808 \pm 0.02	0.816 \pm 0.01	9

Table A-6: Classification accuracy for a 32 unit LSTM model with the dense layer retrained with increasing amount of target data. The table shows the classification accuracy for the target user training, validation and test data sets $\pm\sigma(n = 25)$. A value of one represents 100% correct classification.

(a) Subject 01

Target Windows	Training	Validation	Testing	Average Epochs
100	0.924 \pm 0.02	0.714 \pm 0.03	0.786 \pm 0.03	4
250	0.920 \pm 0.02	0.688 \pm 0.03	0.798 \pm 0.02	4
500	0.924 \pm 0.02	0.731 \pm 0.03	0.820 \pm 0.03	4
750	0.930 \pm 0.02	0.741 \pm 0.03	0.826 \pm 0.02	4
1000	0.929 \pm 0.02	0.771 \pm 0.02	0.843 \pm 0.02	4
1500	0.928 \pm 0.02	0.775 \pm 0.03	0.841 \pm 0.02	5
3000	0.935 \pm 0.02	0.815 \pm 0.01	0.833 \pm 0.01	5
6000	0.945 \pm 0.01	0.788 \pm 0.04	0.840 \pm 0.02	5
9000	0.953 \pm 0.01	0.749 \pm 0.03	0.836 \pm 0.01	6
12000	0.946 \pm 0.01	0.773 \pm 0.02	0.841 \pm 0.02	5
15000	0.916 \pm 0.01	0.804 \pm 0.02	0.867 \pm 0.01	5

(b) Subject 03

Target Windows	Training	Validation	Testing	Average Epochs
100	1.000 \pm 0.00	0.953 \pm 0.02	0.792 \pm 0.03	7
250	1.000 \pm 0.00	0.952 \pm 0.02	0.789 \pm 0.03	6
500	1.000 \pm 0.00	0.972 \pm 0.01	0.805 \pm 0.02	8
750	1.000 \pm 0.00	0.973 \pm 0.01	0.807 \pm 0.02	8
1000	1.000 \pm 0.00	0.976 \pm 0.01	0.819 \pm 0.02	8
1500	1.000 \pm 0.00	0.969 \pm 0.01	0.822 \pm 0.02	7
3000	0.984 \pm 0.00	0.595 \pm 0.06	0.777 \pm 0.05	4
6000	0.975 \pm 0.00	0.726 \pm 0.02	0.856 \pm 0.02	4
9000	0.963 \pm 0.00	0.817 \pm 0.02	0.858 \pm 0.01	4
12000	0.957 \pm 0.01	0.874 \pm 0.02	0.900 \pm 0.01	9
15000	0.945 \pm 0.01	0.826 \pm 0.02	0.896 \pm 0.01	6

(c) Subject 09

Target Windows	Training	Validation	Testing	Average Epochs
100	0.992 \pm 0.00	0.834 \pm 0.06	0.794 \pm 0.03	4
250	0.987 \pm 0.00	0.833 \pm 0.03	0.840 \pm 0.02	4
500	0.985 \pm 0.00	0.851 \pm 0.03	0.843 \pm 0.02	4
750	0.980 \pm 0.00	0.866 \pm 0.02	0.845 \pm 0.02	4
1000	0.979 \pm 0.00	0.870 \pm 0.02	0.853 \pm 0.02	4
1500	0.978 \pm 0.00	0.876 \pm 0.02	0.856 \pm 0.03	5
3000	0.974 \pm 0.01	0.953 \pm 0.02	0.773 \pm 0.04	10
6000	0.968 \pm 0.00	0.804 \pm 0.02	0.802 \pm 0.04	5
9000	0.964 \pm 0.00	0.719 \pm 0.02	0.849 \pm 0.02	4
12000	0.960 \pm 0.00	0.734 \pm 0.02	0.856 \pm 0.02	4
15000	0.953 \pm 0.00	0.814 \pm 0.01	0.833 \pm 0.02	4

A.3 Tables of Model Performance for Amputee Transfer Learning

This section contains data tables for an investigation into model personalisation for an amputee. Improvements were made by additional training of a general LSTM HAR model with varying amounts of target subject data. Tables A-7, A-8, A-9 and A-10 are included in this section.

Table A-7: Table of results for classification performance for a 32 unit LSTM networks trained with varying amount of amputee subject data. The table shows the classification accuracy for the target user training, validation and test data sets $\pm\sigma(n = 25)$. A value of one represents 100% correct classification.

(a) Intact Limb

Target Windows	Training	Validation	Testing	Average Epochs
100	0.905 \pm 0.04	0.036 \pm 0.43	0.428 \pm 0.05	10
250	0.952 \pm 0.01	0.012 \pm 0.45	0.448 \pm 0.05	7
500	0.964 \pm 0.02	0.016 \pm 0.86	0.863 \pm 0.02	9
750	0.984 \pm 0.01	0.013 \pm 0.87	0.869 \pm 0.01	8

(b) Prosthetic Limb

Target Windows	Training	Validation	Testing	Average Epochs
100	0.855 \pm 0.10	0.096 \pm 0.45	0.447 \pm 0.02	8
250	0.878 \pm 0.02	0.017 \pm 0.47	0.469 \pm 0.03	5
500	0.896 \pm 0.04	0.038 \pm 0.77	0.765 \pm 0.04	8
750	0.939 \pm 0.04	0.039 \pm 0.83	0.830 \pm 0.03	9

Table A-8: Table of results for classification performance of 32 unit LSTM networks personalised to an amputee using transfer learning. The table shows the classification accuracy for the target user training, validation and test data sets $\pm\sigma(n = 25)$. A value of one represents 100% correct classification.

(a) Intact Limb				
Target Windows	Training	Validation	Testing	Average Epochs
100	0.997 \pm 0.00	0.570 \pm 0.07	0.779 \pm 0.05	6
250	0.991 \pm 0.01	0.547 \pm 0.07	0.764 \pm 0.07	4
500	0.976 \pm 0.01	0.827 \pm 0.02	0.900 \pm 0.01	5
750	0.981 \pm 0.01	0.832 \pm 0.02	0.898 \pm 0.02	4

(b) Prosthetic Limb				
Target Windows	Training	Validation	Testing	Average Epochs
100	0.962 \pm 0.06	0.443 \pm 0.04	0.727 \pm 0.04	6
250	0.949 \pm 0.07	0.517 \pm 0.04	0.676 \pm 0.04	4
500	0.931 \pm 0.03	0.794 \pm 0.03	0.867 \pm 0.02	5
750	0.959 \pm 0.02	0.818 \pm 0.03	0.866 \pm 0.02	5

Table A-9: Table of results for classification performance of 32 unit LSTM networks with the dense layer personalised to an amputee using transfer learning. The table shows the classification accuracy for the target user training, validation and test data sets $\pm\sigma(n = 25)$. A value of one represents 100% correct classification.

(a) Intact Limb

Target Windows	Training	Validation	Testing	Average Epochs
100	0.932 \pm 0.05	0.539 \pm 0.05	0.776 \pm 0.02	7
250	0.934 \pm 0.02	0.610 \pm 0.03	0.784 \pm 0.02	6
500	0.920 \pm 0.02	0.837 \pm 0.02	0.881 \pm 0.02	10
750	0.930 \pm 0.02	0.869 \pm 0.02	0.889 \pm 0.01	9

(b) Prosthetic Limb

Target Windows	Training	Validation	Testing	Average Epochs
100	0.834 \pm 0.11	0.388 \pm 0.06	0.686 \pm 0.05	7
250	0.812 \pm 0.09	0.541 \pm 0.04	0.703 \pm 0.04	5
500	0.845 \pm 0.08	0.769 \pm 0.04	0.847 \pm 0.07	11
750	0.870 \pm 0.04	0.812 \pm 0.03	0.859 \pm 0.03	11

Table A-10: Table of results for classification performance of 32 unit LSTM networks with the LSTM layer personalised to an amputee using transfer learning. The table shows the classification accuracy for the target user training, validation and test data sets $\pm\sigma$ ($n = 25$). A value of one represents 100% correct classification.

(a) Intact Limb

Target Windows	Training	Validation	Testing	Average Epochs
100	0.978 \pm 0.01	0.574 \pm 0.07	0.813 \pm 0.05	7
250	0.985 \pm 0.01	0.564 \pm 0.09	0.788 \pm 0.07	4
500	0.971 \pm 0.01	0.832 \pm 0.03	0.893 \pm 0.02	7
750	0.973 \pm 0.01	0.835 \pm 0.02	0.892 \pm 0.02	5

(b) Prosthetic Limb

Target Windows	Training	Validation	Testing	Average Epochs
100	0.961 \pm 0.04	0.442 \pm 0.04	0.735 \pm 0.05	8
250	0.929 \pm 0.07	0.537 \pm 0.02	0.707 \pm 0.05	5
500	0.923 \pm 0.02	0.804 \pm 0.02	0.855 \pm 0.02	6
750	0.940 \pm 0.02	0.830 \pm 0.02	0.858 \pm 0.02	6

Appendix B

Ethics Forms

This appendix contains the ethics approval received for this study. This received approval from University of Bath Research Ethics Approval Committee for Health (REACH) with reference EP 19/20 003. The ethics forms are presented as submitted.

RESEARCH ETHICS APPROVAL COMMITTEE FOR HEALTH

Application form for full submission for research ethics approval

Staff		PhD	x	Masters		UG		Other (e.g. MRes)	
-------	--	-----	---	---------	--	----	--	-------------------	--

ESRC funded project or studentship	x	Knowledge Transfer Partnership		Consultancy	
Other funded or unfunded research project		Service evaluation/Audit		Other (Umbrella etc. please specify)	

Project Title	Deep learning for environmental state prediction and sensor fusion for intelligent wearable robots
Name of applicant/s	Frederick Sherratt
Email for applicant/s	Fs349@bath.ac.uk
Name & contact email for supervisor (for UG / Masters / PhD students)	Pejman Iravani Pi304@bath.ac.uk
Department	Mechanical Engineering
Proposed dates of study	Start: 10-02-2020 End: 19-05-2020

Secondary data analysis
<p>Does this proposal involve secondary data analysis? This is when you are analysing data that has already been collected by somebody else, i.e. you will have no part in collecting the original data.</p> <p style="text-align: center;">YES <input type="checkbox"/> NO <input type="checkbox"/></p> <p>N.B. Please attach evidence that of ethical approval for the original study. The Project Description should detail what you intend to do with the data, not how the data were originally collected. It is important to note whether the data you are using have already been anonymised.</p>

<i>Are there ethical implications concerned with the following general issues? If yes, please provide details below</i>	
1. Funding source	No – EPSRC funded studentship
2. Freedom to publish the results	No – EPSRC funded studentship
3. Future use of findings	No
4. Conflicts of Interest	No

<p>Information Classification Scheme</p> <p>Confirm that you have completed the mandatory information security awareness online training module (available here: https://moodle.bath.ac.uk/course/view.php?id=56392)</p> <p>What category of data will you be collecting? (If you are unsure, please look at the guidance available on the REACH wiki)</p> <p>Internal Use Restricted Highly Restricted</p>
--

<p>Standard Operating Procedures (SOPs)</p> <p>This link will take you to the SOPs for the Department for Health:</p> <p>http://www.bath.ac.uk/health/research/research-getting-started/sop.bho/index.html</p> <p>The SOP PDFs can be downloaded and printed, but before using a printed SOP please check this link to make sure you have the most up to date version. The SOP for the creation and approval of all other SOPs can be found here:</p> <p>http://www.bath.ac.uk/health/research/research-getting-started/sop.bho/SOP_PDFs/SOPs_SOP_V4.pdf</p>

DESCRIPTION OF RESEARCH

1 Research Title	Deep learning for environmental state prediction and sensor fusion for intelligent wearable robots
2 Background and aims of the research (no more than 300 words)	<p>It is estimated that 150,000 people per year will suffer a below knee amputation and this is increasing due to the rising prevalence of diabetes and related vascular disease. The loss of mobility that comes from loss of limb leads to decreased social and economic participation as well as further health issues. Advanced lower limb prosthetics are so far unable to adequately emulate lost muscle function.</p> <p>For lower prosthesis to emulate the lost limb, they must behave in a predictable manner that matches the users intent. The study will investigate the use of machine learning techniques in achieving this. Primarily it will look at mode selection based on time series sensor data.</p>

<p>3 Outline the study design and list the methods including any questionnaires/interview schedules (please attach).</p> <p>How much time (roughly) will each method take and how long in total will participants be expected to take part in the study (maximum 300 words)</p>	<p>This study will produce a large set of labelled activity data from able-bodied subjects is required. The data sets will be raw IMU (Inertial Measurement Unit) sensor readings along with contextual statistics on sample population Age and Sex. The data set will be fully anonymised.</p> <p>Participants will be required for one 30 minutes session of; 15 minutes for briefing, consent and setup, 15 minutes for data collection. During data collection, participants will be required to perform ambulation around a natural built-up environment. The route will involve flats, stairs and ramps with participants guided by a researcher.</p> <p>Participants will wear a set of seven Movesense wireless IMU sensors. The Movesense is a watch sized device containing a 9 axis IMU providing accelerometers, gyroscopes and magnetometers readings. Data will be streamed over Bluetooth to a smartphone controlled by the researcher.</p> <p>The Movesense devices will be attached to the ankle, wrist, hip and chest using Velcro/elastic straps and clothes clips. This will be done in a non-invasive manner to minimise discomfort and its potential effects on movement.</p> <p>Sensors will be controller via the smartphone. The researcher will annotate activity of the incoming data using a custom smartphone app.</p>
<p>4 Who will be recruited to participate in the research?</p>	<p>A broad population is required to ensure adequate variation in the dataset. Adult students and staff member of the University population will be recruited.</p>
<p>5 How many participants will be recruited? Why is this number necessary?</p>	<p>The sample size will be from 25 to 50 participants. Machine learning requires large amounts of data set to produce a generalised solution. The larger the number of participants the higher quality the model.</p>
<p>6 How will participants be recruited?</p>	<p>Participants will be recruited from the general university population through personal contacts from the engineering department and sports teams. Care will be taken to ensure adequate variety in participants.</p>
<p>7 Are there any potential participants who will be excluded? If so, what are the exclusion criteria? Is there any specific inclusion criteria?</p>	<p>Underage participants and those unable to perform the required activities unaided will be excluded. All participants must be able to provide written consent. There are no further inclusion/exclusion criteria.</p> <p>Exclusion/inclusion from the study will be determined by participant self-assessment as part of the brief and consent.</p>
<p>8 Where will the research take place?</p>	<p>Department of Mechanical Engineering and University of Bath Campus</p>

<p>9 How will informed consent be obtained from all participants or their parents/guardians prior to individuals entering the study?</p>	<p>All participants will be required to read the full participant information sheet. Following verbal explanation of the aims, methods, objectives and potential risks of the study, written informed consent will be obtained from each volunteer. It will be made clear that participants are free to deny consent (withdraw) before, during, or immediately after data collection.</p>
<p>10 If the study aims to actively deceive the participants, please justify and briefly outline how this will be carried out</p>	<p>-</p>
<p>11 Will participants be made aware they can drop out of the research study at any time without having to give a reason for doing so?</p> <p>Is it clear at what point participants can withdraw their data (e.g. before anonymization)?</p>	<p>Yes, participants can drop out at any point during or immediately after data collection. Data will be anonymised immediately after the recording session.</p>
<p>12 Describe any potential risks to participants (physical, psychological, legal, social) arising from the study. Explain how you will seek to resolve these.</p>	<p>None, the study only requires participants to perform natural movements in a public environment. Sensors will be attached in a non-invasive, non-intimate manner.</p>
<p>13 Describe any potential benefits of the study for the participants</p>	<p>There are no immediate benefits to participation. The results of this project will generate a dataset from which to exploit future research opportunities in the field of amputee biomechanics.</p>
<p>14 Describe potential risks to researcher/s and how these will be managed.</p>	<p>None, all data collection will be performed in a public environment.</p>
<p>15 How will participants be debriefed? (i.e. feedback of results)</p> <p>What aftercare will you provide?</p>	<p>No debrief is required. No aftercare is required.</p>
<p>16 How will confidentiality and security of personal data relating to your participants be maintained?</p>	<p>Data is anonymized at collection. Personal information about the participants (Age, Sex) will not be linked to the anonymised data and only aggregated statistics will be published. All data will be stored in a restricted access folder on the X drive.</p>

17 Will the participants be photographed, audio-taped or video-taped? If so, please justify	No audio/video is not required for the study. With participants permission photographs may be taken for use in subsequent reports. Participant anonymity will be maintained in any publication.
18 Is any reimbursement of expenses or other payment to be made to participants? Please explain.	None – it will be made clear this is a voluntary study.
19 Any other relevant information?	-
20 How long will you store <i>personal</i> data (including informed consent)? If you are retaining personal data longer than the end of the study, please justify	Once collected the data will be immediately anonymised. Consent forms will be held until the end of the study.

Attach the following (where relevant) <u>including version number and date</u>:			
		Version	Date
1	Participant information sheets	V2	17/01/2020
2	Consent forms	V2.1	10/02/2020
3	Health history questionnaire	-	
4	Poster/promotional material	-	
5	Debrief	-	
6	Copy of questionnaire/ proposed data collection tool (questionnaire; interview schedule/ observation chart/ data record sheet/ participant record sheet)	-	
7	Data management plan	V1	18/12/2019

Signed by: Principal Investigator or Student Supervisor

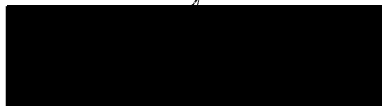


Date: 10/02/2020

By signing and submitting the form, you are agreeing with the following statement:

'I am familiar with the guidelines for ethical practices in research and I have discussed the ethical aspects of the proposed project with my supervisor(s) and/or the other researchers involved in the project. **I am also aware of and will comply with the university policies for storage and processing of human participant data.**'

Signed by: Student or other researchers



Date: 10/02/2020

By signing you are agreeing that you take joint responsibility for the application and conduct of the research.

PARTICIPANT INFORMATION SHEET

Deep learning for environmental state prediction and sensor fusion for intelligent wearable robots

Name of Researcher: Freddie Sherratt (PhD Student in Mechanical Engineering)
Contact details of Researcher: F.W.Sherratt@bath.ac.uk

Name of Supervisor: Dr Pejman Iravani (Senior Lecturer in Mechanical Engineering)
Contact details of Supervisor: P.Iravani@bath.ac.uk
Tel: +44 (0) 1225 384494

This information sheet forms part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. Please read this information sheet carefully and ask one of the researchers named above if you are not clear about any details of the project.

1. What is the purpose of the project:

This research is being carried out as part of PhD research looking into improvements in active lower-limb prosthetic control. This project aims to identify user intent from non-invasive wearable sensor data to enhance locomotion-mode selection.

The loss of mobility that comes from loss of limb leads to decreased social and economic participation as well as further health issues. Current lower-limb prosthetic do not adequately address lost muscle function leading to reduced mobility; falls and fractures; and damage to the spine and remaining leg due to asymmetric and high loading. Active prosthesis have the potential to restore near-natural behaviour but they must behave in a predictable manner that matches the users intent. The study will investigate the use of machine learning techniques in achieving this. Primarily it will look at locomotion-mode selection based on time series sensor data.

2. Who can be a participant?

A broad population is required to ensure adequate variation in the dataset so there are few restrictions on who can participate. The following criteria apply; you must be an adult and able to perform the activities listed in section 4 unaided.

3. Do I have to take part?

Participation in this study is entirely voluntary. Before you decide to take part we will describe the project and go through this information sheet with you. Please feel free to ask any questions you have about the procedures used in the study at any time.

If you agree to take part, we will then ask you to sign a consent form. However if at any time you decide you no longer wish to take part in this project you are free to

withdraw, without giving a reason. Your rights to withdraw shall be preserved over and above the goals of the study.

4. What will I be asked to do?

You will be asked to attend one 45 minute recording session. This will include i) a 15-minute brief and setup, ii) 2 x 10-minute recording session.

Participants will wear a set of seven Movesense wireless IMU sensors. See Figure 1. The Movesense device is a watch-sized device containing a 9 axis Inertial Measurement Unit and Heart rate sensor. Sensor data will be transmitted wirelessly over Bluetooth to a smartphone-controlled by the researcher.



Figure1: Movesense device [www.wearabletechnologyinsights.com/articles/13101/spotlight-on-suunto accessed: 18/12/2019]

The Movesense devices will be attached to the ankle, wrist, hip and chest using Velcro/elastic straps and clothes clips. This will be done in a non-invasive manner to minimise discomfort and its potential effects on movement.

The recording will be in two continuous sessions of approximately 15 minutes. There will be a break between sessions. A typical recording session will be a walk around a pre-set route that includes the following environments:

- Flat indoor and outdoor paved surfaces
- Walking uphill and downhill
- Ascending and descending stairs
- Gentle and abrupt stopping

5. What are the exclusion criteria?

(are there reasons why I should not take part)?

Underage participants and those unable to perform the required activities unaided will be excluded. There are no further inclusion/exclusion criteria.

If you are comfortable performing the previously mentioned activities, there is no reason not to partake.

6. What are the possible benefits of taking part?

There are no immediate benefits to participation. You will be aiding in the generation of a new data set to aid research looking into Locomotion Mode Recognition.

7. What are the possible disadvantages and risks of taking part?

There are no disadvantages to you taking part in the project.

- The experimental trials will not require you to perform any other activity outside of those listed with all activities will be undertaken at your own pace.
- All measurements taken will be non-invasive and any attached instrumentation (Movesense wireless IMUs and associated attachments) will be external and non-obtrusive.
- Any testing session will stop should you report, or appear to be unduly stressed.

8. Will my participation involve any discomfort or embarrassment?

We do not expect you to feel any discomfort or embarrassment if you take part in the project. Testing sessions will be stopped should you wish too, or if you appear unduly stressed.

The experimental trials will not require you to perform any other activity outside of the required ones. All required sensors are designed to be comfortable for extended periods and will be attached external and non-obtrusive.

9. Who will have access to the information that I provide?

Only the research team will have access to the personal information that you provide. A research data set will be published in an anonymous form as per the University's Code of Practice – Research. This will be retained for a minimum of 10 years.

10. What will happen to the data collected and results of the project?

Data is collected in an anonymous and will be stored on password-protected PCs and only accessed by researchers directly involved in the study. There will be no links between persons involved in the study and the data they generate.

Participant anonymity will be maintained in any publication. In any manuscripts, reports or publications resulting from this study codes rather than names will be used. Any images published will not include faces.

Upon study completion, the anonymised data will be made publically available to aid other researchers in the field. This will be stored for a minimum of 10 years to comply with the University's Code of Practice – Research. No personal data will be released.

11. Who has reviewed the project?

This project has been given a favourable opinion by the University of Bath, Research Ethics Approval Committee for Health (REACH) [reference: EP 19/20 003].

12. How can I withdraw from the project?

You are free to deny consent (withdraw) before, during or immediately after data collection. Once data has been anonymized you will no longer be able to withdraw from the study.

Your rights to withdraw shall be preserved over and above the goals of the study.

13. University of Bath privacy notice

The University of Bath privacy notice can be found here:

<https://www.bath.ac.uk/corporate-information/university-of-bath-privacy-notice-for-research-participants/>.

14. What happens if there is a problem?

If you have a concern about any aspect of the project you should ask to speak to the researchers who will do their best to answer any questions.

If they are unable to resolve your concern or you wish to make a complaint regarding the project, please contact the Chair of the Research Ethics Approval Committee for Health:

Professor James Betts
Email: j.betts@bath.ac.uk
Tel: +44 (0) 1225 383448

15. If I require further information who should I contact and how?

Thank you for expressing an interest in participating in this project. Please do not hesitate to get in touch with us if you would like some more information.

Name of Researcher: Freddie Sherratt (PhD Student in Mechanical Engineering)
Contact details of Researcher: F.W.Sherratt@bath.ac.uk

Name of Supervisor: Dr Pejman Iravani (Senior Lecturer in Mechanical Engineering)
Contact details of Supervisor: P.Iravani@bath.ac.uk
Tel: +44 (0) 1225 384494

CONSENT FORM

Deep learning for environmental state prediction and sensor fusion for intelligent wearable robots

Name of Researcher: Freddie Sherratt (PhD Student in Mechanical Engineering)
Contact details of Researcher: F.W.Sherratt@bath.ac.uk

Name of Supervisor: Dr Pejman Iravani (Senior Lecturer in Mechanical Engineering)
Contact details of Supervisor: P.Iravani@bath.ac.uk
Tel: +44 (0) 1225 384494

Please initial each box if you agree with the statement

1. I have been provided with information explaining what participation in this project involves.
 2. I have had an opportunity to ask questions and discuss this project.
 3. I have received satisfactory answers to all questions I have asked.
 4. I have received enough information about the project to make a decision about my participation.
 5. I understand that I am free to withdraw my consent to participate in the project at any time without having to give a reason for withdrawing.
 6. I understand that I can withdraw from the study at any time (up until immediately after the completion of the testing session).
 7. I understand the nature and purpose of the procedures involved in this project. These have been communicated to me on the information sheet accompanying this form.
 8. I understand that the University of Bath may use the data collected for this study in future research project(s) but that the conditions on this form under which I have provided the data will still apply
 9. I understand the data I provide will be treated as confidential, and that on completion of the project my name or other identifying information will not be disclosed in any presentation or publication of the research.
 10. I agree to the University of Bath keeping and processing the data that I provide during the course of this study and my consent is conditional upon the University complying with its duties and obligations under the Data Protection Act
-
11. I am capable of performing the required activities, described in the Participant Information Sheet, unaided and for the required duration
 12. I understand that all activities will take place at my own pace and can be stopped at any time if requested

13. I give consent for the research team to take photographs of my person for written or oral presentations such as journal articles, conference presentation and reports. This is not a requirement to participate in the study.

14. I understand that in any published photographs my anonymity will be preserved at all times.

15. I hereby fully and freely consent to my participation in this project.

Participant's signature: _____ Date: _____

Participant name in BLOCK Letters: _____

Researcher's signature: _____ Date: _____

Researcher name in BLOCK Letters: _____

If you have any concerns or complaints related to your participation in this project please direct them to the Chair of the Research Ethics Approval Committee for Health, Dr James Betts (j.betts@bath.ac.uk, 01225 383448)

Doctoral Data Management Plan Template

1 Overview

1.1 Project title
Deep learning for environmental state prediction and sensor fusion for intelligent wearable robots
1.2 Student name and department
Frederick Sherratt – Mechanical Engineering
1.3 Supervisor(s)
Note: the main University of Bath supervisor is the Data Steward for the project. Dr Pejman Iravani* – Department of Mechanical Engineering Prof Andrew Plummer – Department of Mechanical Engineering
1.4 Project description
The project aims to determine if machine-learning techniques can be used to improve performance of lower-limb prosthetics. This research focuses on locomotion-mode selection and as such a labelled data set of different locomotion-modes and transitions between is required.

2 Compliance

When you submit your DMP you are confirming that you have read and understood all of the legislative, policy and contractual requirements that apply to your project.

Information on additional University of Bath policies and UK/EU legislation that may apply to research can be found in our [Data Management Plan Compliance Wiki page](#) (this will require you to sign in with your University of Bath user account).

2.1 University policy requirements
Data must be stored securely in a manner that minimises the risk of loss of data and licenced in the as open a manner as possible. Sufficient meta data must be provided to allow others to use the data. All publication must include a data access statement.
University policy or guidance
University of Bath Research Data Policy
University of Bath Code of Good Practice in Research Integrity
University of Bath Electronic Information Systems Security Policy
University of Bath Intellectual Property Policy
University of Bath Code of Ethics
2.2 Legal requirements
Data collected from human subjects must comply with personal data protection regulations. Informed consent must be obtained from participants for data to be retained, shared, and used for new purposes.
UK Legislation or framework
General Data Protection Regulations



2.3 Contractual requirements	
EPSRC funding requires that all publicly funded research must publish any data collected into the public domain.	
Name of funder	Data policy URL
EPSRC	https://epsrc.ukri.org/about/standards/researchdata/

3 Gathering data

There is guidance and example wording for this section on the [Data Management Plan Guidance Wiki page](#).

3.1 Description of the data
3.1.1 Types of data The data will be raw accelerometer, gyroscope and magnetometer readings from seven IMUs along with timestamps of each sample recording, a set of timestamped activity labels is, and participant consent forms
3.1.2 Format and scale of the data Original data will be in the form of .txt files containing hexadecimal representations of int16 fixed point numbers and associated uint32 millisecond timestamps. A second set of .txt files will contain millisecond timestamps and activity labels. A converted human-readable .csv form of this data will also be produced. Data will be less than 100Mb per participant. Personal data will be paper consent forms and Sex and Gender information for each participant. These will have no link to the raw data files.
3.2 Data collection methods
Data will be collected using an android smartphone app that connects to the Movesense Bluetooth IMU sensors. The smartphone will save the data in its raw form to its internal storage.
3.3 Development of original software
Original software is required for all stages of data collection and processing. Embedded C++ software has been developed to run on the Movesense devices transmitting their raw IMU data. A java Android app has been developed to connect to the sensors and save the raw data and activity labels to .txt files. Matlab scripts have been developed to convert the encoded raw data back to a human readable form. The software has all developed specifically for this project and version controlled using Github. Associated documentation for their use is provided with the programs.

4 Working with data

There is guidance and example wording for this section on the [Data Management Plan Guidance Wiki page](#).

4.1 Short- and medium-term data storage arrangements
In the short to medium term data will be stored in a X drive folder. All original data will be stored in a read-only format to prevent accidental overwrite.

Consent forms will be stored in a lock filing cupboard in my office until the end of the project when they will be securely destroyed.

4.2 Control of access to data and sharing with collaborators

The X drive folder will have access control restricted to myself and my research group.

4.3 File organisation and version control

Each participant's data will be stored in a separate folder with a random unique identifier to anonymise the data.

4.4 Documentation that will accompany the data

A readme file will accompany the data set explaining how to interpret the data and providing context.

5 Archiving data

There is guidance and example wording for this section on the [Data Management Plan Guidance Wiki page](#).

5.1 Selection of data to be retained and deleted at the end of the project

Only processed data and aggregated population statistics will be retained after the end of the project. The original raw data and all consent forms will be destroyed

5.2 Data preservation strategy and retention period

As with other secondary data sources in the field data will be published in the University's Research Data Archive for a minimum of ten years.

5.3 Maintenance of original software

Scripts and software will not be maintained beyond the end of this project. Software will be archived with appropriate documentation for use again.

6 Sharing data

There is guidance and example wording for this section on the [Data Management Plan Guidance Wiki page](#).

6.1 Justification for any restrictions on data sharing

No personal data will be included in published data set therefore there are no restrictions on data sharing.

6.2 Arrangements for data sharing

Data will be shared through the universities research archive with a data access statement provided in any publication based on it.

7 Implementation

There is guidance and example wording for this section on the [Data Management Plan Guidance Wiki page](#).

**7.1 Review of the Data Management Plan**

The data management plan will be reviewed at the end of this experiment phase and before any subsequent experiments.

7.2 Special resources required for the project

No special resources required

7.3 Further training needs

I have attended data management and archiving courses and currently do not require further training.

Appendix C

Sensor Datasheet

This appendix contains the data sheet for the Movesense wireless IMU used during data collection[219].

MOVESENSE SENSOR

Versatile, light and small but extremely durable sensor capable of measuring any movement and much more. Customizable functionality through open APIs that enable development of unique in-device apps. The functionality can be tailored to fit the exact needs of the target use case.

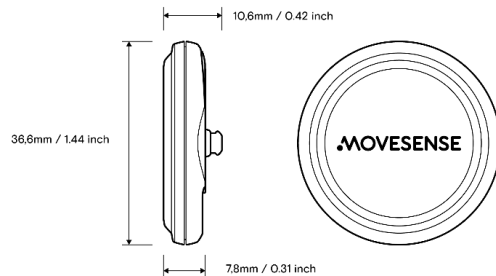


- Swim and shock proof construction, suitable for any sports
- Low profile snap connection for smooth and subtle attachment to apparel or gear
- User replaceable coin cell battery
- State of the art ultra-low power components
- Small size, light weight and waterproof
- Based on Suunto design and development
- Developed, designed and manufactured in Finland
- Available with custom branding

TECHNICAL HIGHLIGHTS

- 9-axis movement sensor: acceleration, gyroscope, magnetometer
- Heart rate, R-R- intervals, BLE heart rate service, optional: single channel ECG
- 1-wire expansion bus
- Temperature
- Data logging memory
- Bluetooth® 4.0 / 5.0 radio depending on firmware version
- Tools for developing customized applications that run inside the sensor
- Software libraries for developing compatible mobile applications
- Wireless firmware update capability
- Recognizes its attachment base through unique ID in Movesense connector

TECHNICAL BRIEF



DIMENSIONS

- 36.6mm/1.44" dia. x 10.6mm/0.42" thick
- Weight 9.4g / 0.33oz with battery
- Water resistant to 30m/100ft

I/O

- Red led on the front, SW controllable
- Wake-up, heart rate and 1-wire expansion
- Interface via Movesense studs, stud center-to-center distance: 27.0mm/1.06"
- Optional attaching accessories with unique ID (readable through 1-wire)

SENSORS

- Accelerometer & Gyroscope
 - $\pm 2/\pm 4/\pm 8/\pm 16g$ full scale
 $\pm 125/\pm 245/\pm 500/\pm 1000/\pm 2000^\circ/s$ full scale, sampling frequency: 12.5/26/52/104/208Hz/416Hz/833Hz
- Magnetometer ± 49 gauss full scale
- Temperature
 - Accuracy $<\pm 0.5^\circ C$, $0^\circ C$ to $+65^\circ C$
- Heart rate
 - Beats/minute, RR intervals, BLE HR service
 - 1 Channel ECG (non-medical)

LOGGER MEMORY

- 3Mbit EEPROM

SOFTWARE

- SDK for developing apps for the sensor
- Sensors and peripherals controllable via API incl. BT advertising, power schemes
- Easy to use C++ Movesense Device API
- iOS and Android mobile libraries with wireless
- sensor firmware update capability
- GNU toolchain for embedded ARM

MCU

- Nordic Semiconductor nRF52832
- 32-bit ARM® Cortex®-M4
- 64kB on-chip RAM*
- 512kB on-chip FLASH*
- (*) Memory is shared with the Movesense OS and the user application
- Bluetooth Low Energy radio

BATTERY

- CR 2025 Lithium coin cell battery
- Operating time up to months, depending on the user application

APPROVALS AND COMPLIANCES

- CE, FCC, IC, C-Tick, CMIIT
- Conforms REACH, RoHS
- Bluetooth 4.0 / 5.0

PATENTS

US 13/071,624, US 13/832,049, US 13/832,598, US 13/917,668, US 13/397,872, USD 667,127, US 8,386,009, US 8,750,959, US 8,814,574, US 8,886,281, others pending

Appendix D

Sample Data

This chapter contains full summary tables for the first hand data collected (Section D.1) and plots of sample data (Section D.2).

D.1 Summary Tables of Collected Data

Table D-1 contains the demographic data for all test subjects. Table D-2 contains a summary of the data collected during the first phase of data collection. Table D-3 show a summary of the data collected during the second phase. Table D-4 contains a summary of the first hand data collected during for one left trans-tibial individual.

Table D-1: Table of demographic data for study subjects

Subject ID	Age	Gender	Height [cm]	Weight [Kg]
01	27	Male	185	75
02	24	Male	180	77
03	25	Male	180	80
04	23	Male	180	72
05	24	Male	178	82
06	23	Male	170	68
07	24	Male	172	68
08	24	Male	165	80
09	26	Female	173	65
10	23	Male	180	75
11	26	Male	185	75
12	26	Female	170	68
13	45	Male	175	80
14	26	Male	182	85
15	25	Male	180	75
16	26	Male	187	80
17	25	Female	161	60
18	23	Female	163	63
19	24	Female	165	65
20	26	Male	186	90
21	63	Female	165	75
22	25	Male	185	70
A1	56	Male	178	70

Table D-2: Summary of non-amputee data collected during the first phase of collection. (Ramp Ascent (RA), Ramp Descent (RD), Stair Ascent (SA), Stair Descent (SD))

Subject ID	WALK	RA	RD	SA	SD	STOP
01	34618	0	0	5349	5025	1281
02	13564	0	0	4614	4447	1020
03	5005	0	0	3633	2568	1587
04	32336	0	0	6045	5482	0
05	32525	0	0	6621	5767	0
06	37895	0	0	8424	5636	0
07	21843	0	0	14074	10873	0
08	38590	0	0	11036	17801	0
09	40384	0	0	10826	7953	0
10	37353	0	0	10812	8084	0
11	8341	0	0	1614	1566	0
12	9038	0	0	6273	4926	0
13	252022	48821	53735	18778	16220	16887
14	302440	18531	18305	10936	9581	73781
15	12249	0	0	1452	1929	3498
16	23729	0	0	5332	2578	1651
17	113222	2702	3754	3190	3949	19127
18	37352	0	0	3747	2565	1133
19	4990	0	0	1240	1245	0
20	3487	0	0	2383	3054	0
21	4806	0	0	2551	190	206
22	9033	3274	3630	982	938	856
Total	1075111	73328	79426	139915	122378	121027

Table D-3: Data samples of non-amputee data collected during the second phase of collection. (Ramp Ascent (RA), Ramp Descent (RD), Stair Ascent (SA), Stair Descent (SD))

Subject ID	WALK	RA	RD	SA	SD	STOP
01	462446	141268	139786	59685	44024	62397
03	291213	77508	59157	48695	50210	157867
09	368090	115299	82980	49530	51698	60605
Total	2100308	404127	364574	277250	252494	363669

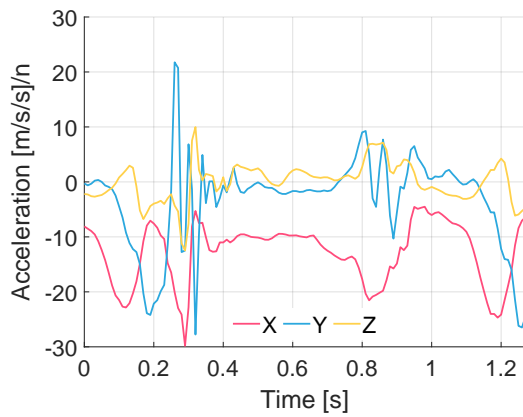
Table D-4: Data samples of first hand amputee data collected during the third phase of collection. (Ramp Ascent (RA), Ramp Descent (RD), Stair Ascent (SA), Stair Descent (SD))

Subject ID	WALK	RA	RD	SA	SD	STOP
A1	38114	6159	7194	2872	2450	11763

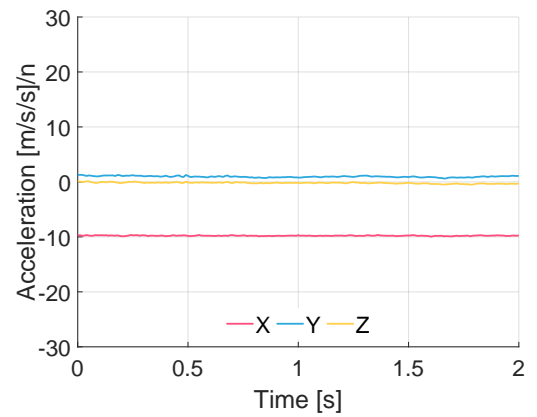
D.2 Example Sensor Data

Within this section example plots of the recorded sensor data are provided for reference. The accelerometer and gyroscope data for each of the five sensor locations are included. Each Figure shows example data for each of the six activities labelled.

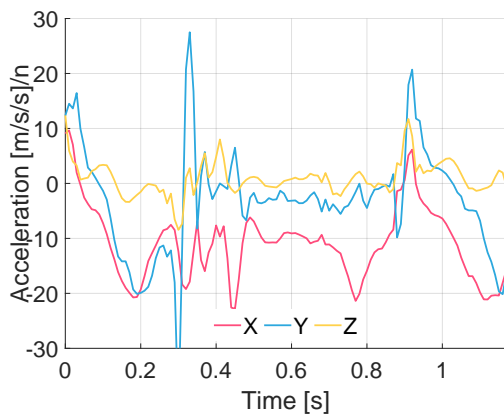
Figures D-1 and D-2 show example data for the left ankle accelerometer and gyroscope respectively. Figures D-3 and D-4 the right ankle. Figures D-5 and D-6 show the left hip and Figures D-7 and D-8 the right hip. Figures D-9 and D-10 contain example plots for the chest accelerometer and gyroscope.



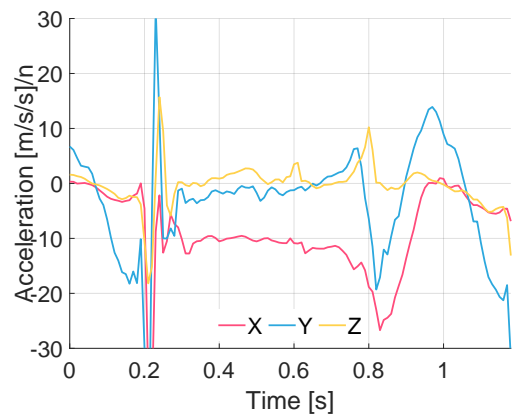
(a) Walking



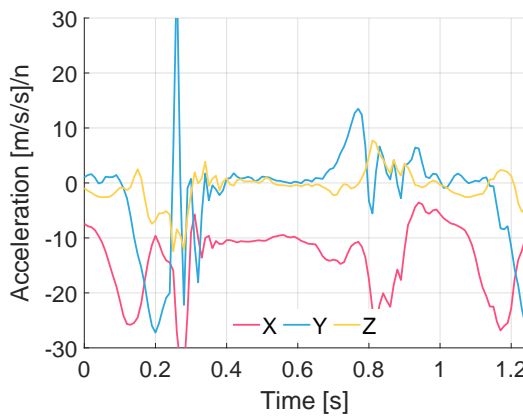
(b) Stopped



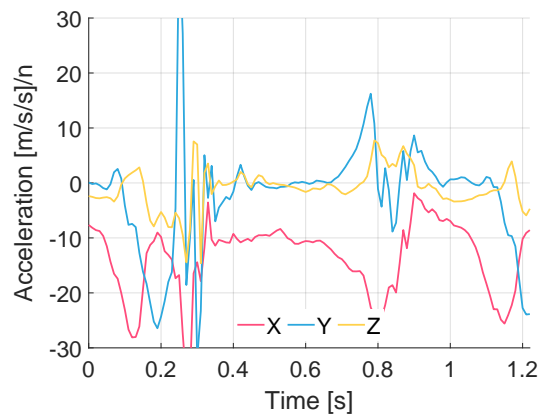
(c) Stair Ascent



(d) Stair Descent

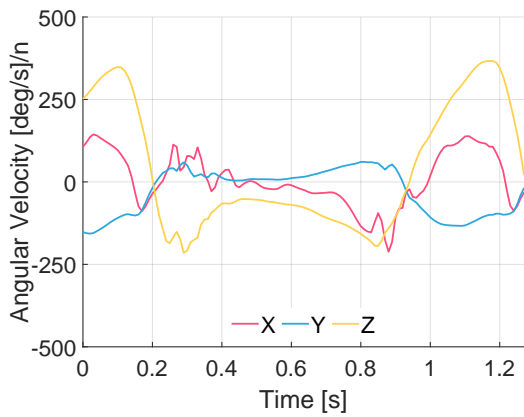


(e) Ramp Ascent

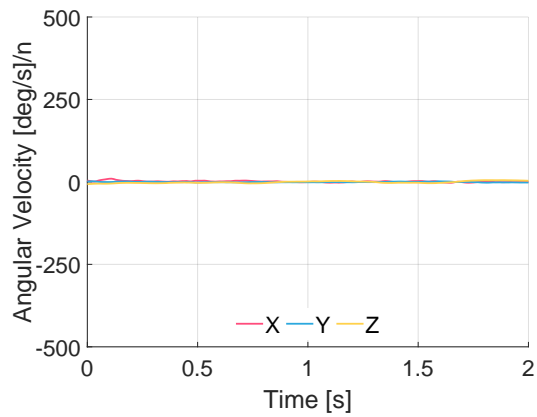


(f) Ramp Descent

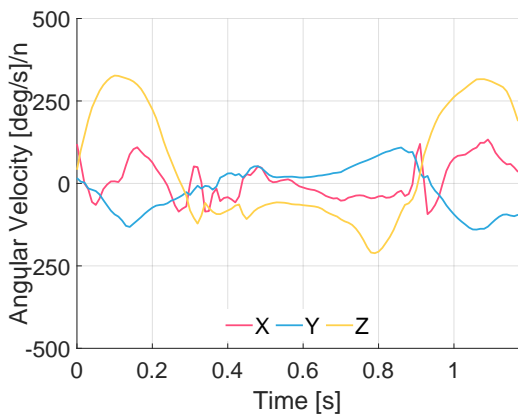
Figure D-1: Example data for the left ankle accelerometer. The x represent recording time in seconds. The y axis show the measured acceleration in $m/s/s$. The red lines represents the x axis of the sensor, the blue solid lines the y axis and the yellow lines the z axis.



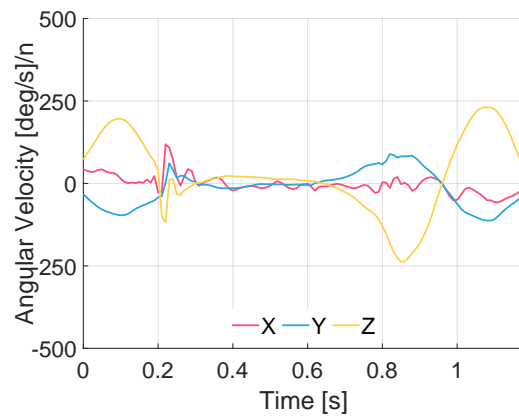
(a) Walking



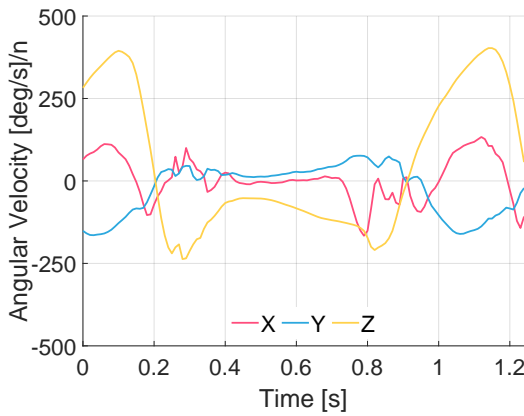
(b) Stopped



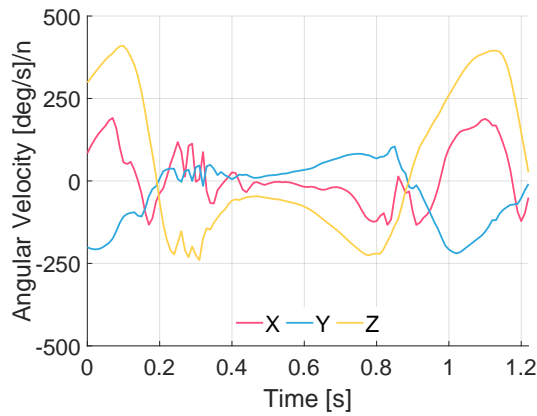
(c) Stair Ascent



(d) Stair Descent

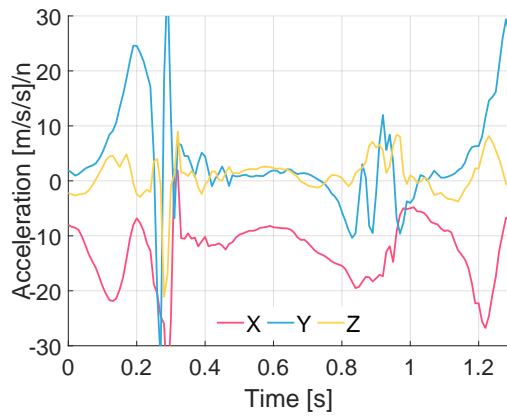


(e) Ramp Ascent

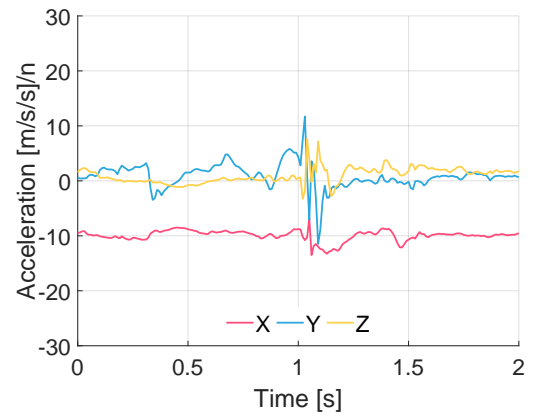


(f) Ramp Descent

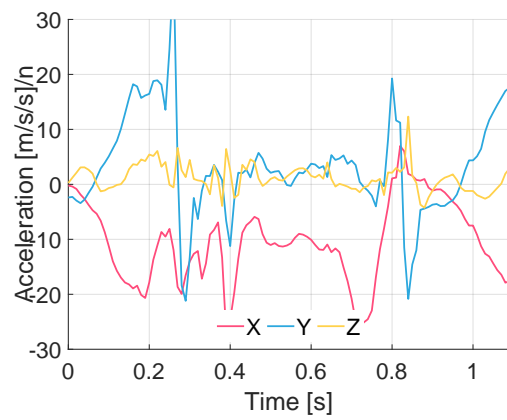
Figure D-2: Example data for the left ankle gyroscope. The x represent recording time in seconds. The y axis show the measured angular velocity in deg/s. The red lines represents the x axis of the sensor, the blue solid lines the y axis and the yellow lines the z axis.



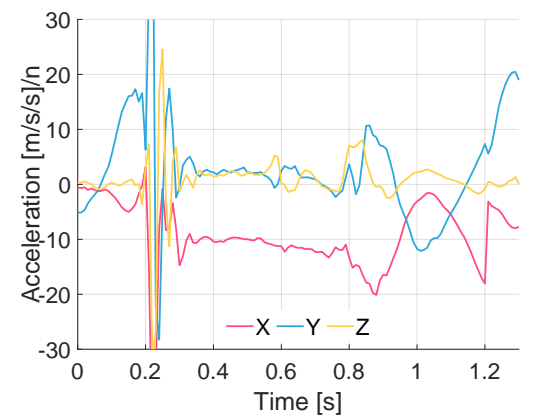
(a) Walking



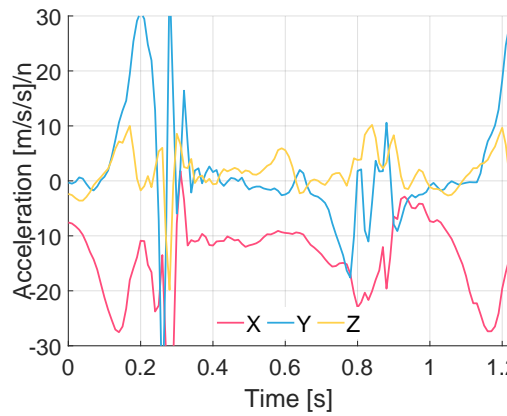
(b) Stopped



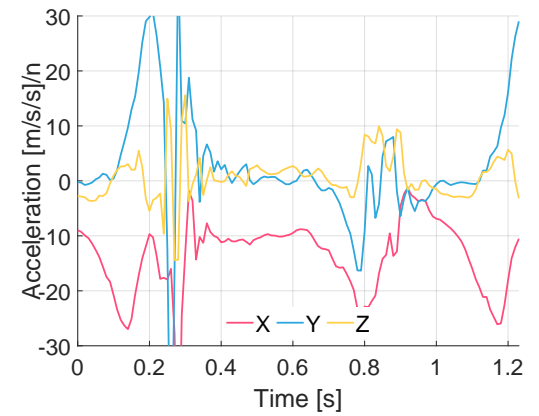
(c) Stair Ascent



(d) Stair Descent

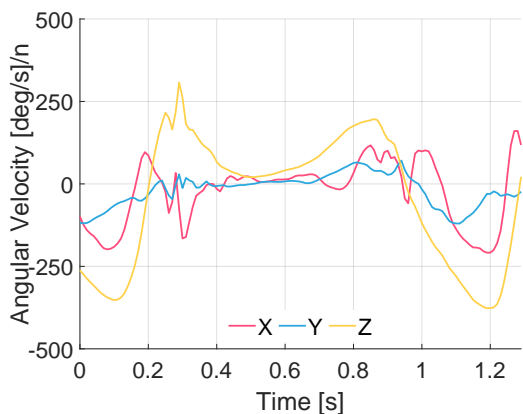


(e) Ramp Ascent

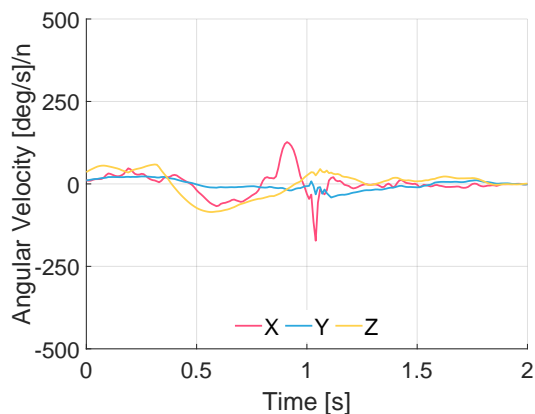


(f) Ramp Descent

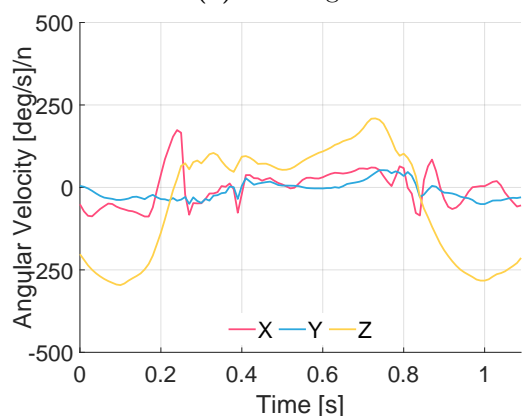
Figure D-3: Example data for the right ankle accelerometer. The x represent recording time in seconds. The y axis show the measured acceleration in $m/s/s$. The red lines represents the x axis of the sensor, the blue solid lines the y axis and the yellow lines the z axis.



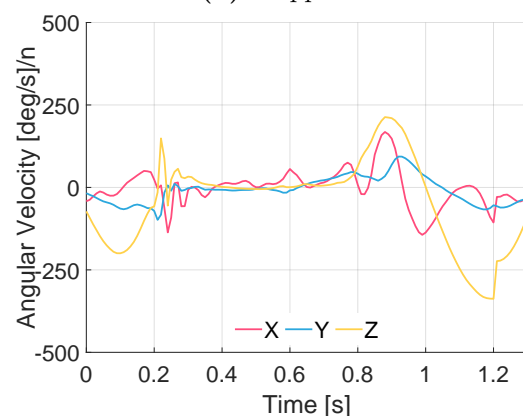
(a) Walking



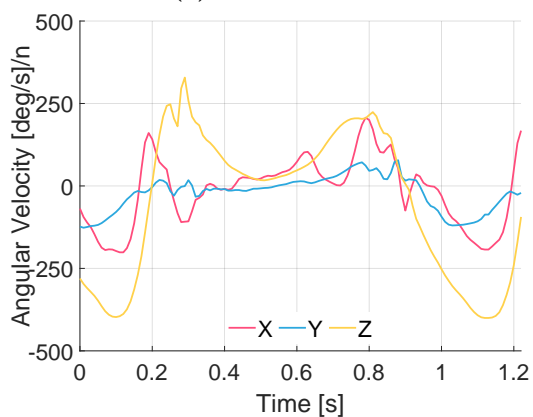
(b) Stopped



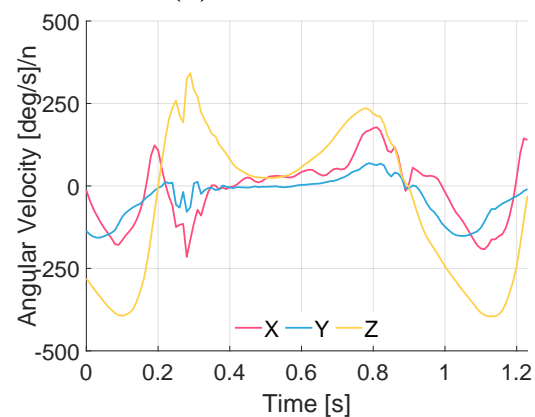
(c) Stair Ascent



(d) Stair Descent

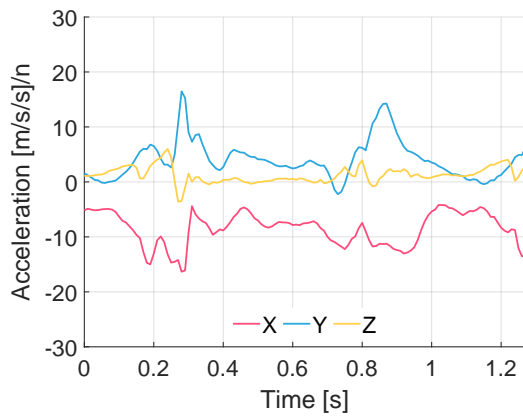


(e) Ramp Ascent

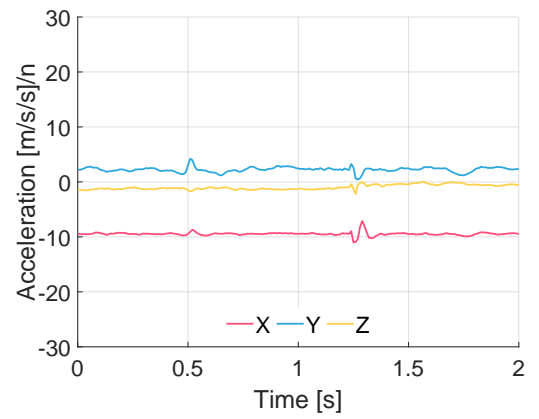


(f) Ramp Descent

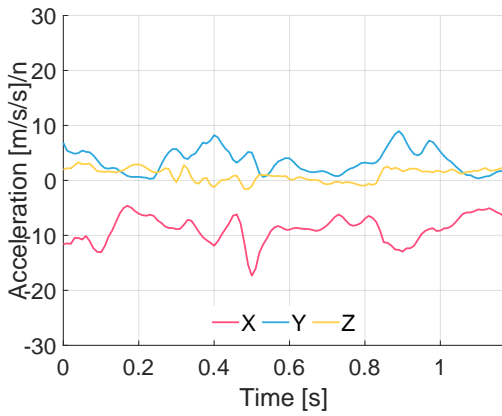
Figure D-4: Example data for the right ankle gyroscope. The x represent recording time in seconds. The y axis show the measured angular velocity in deg/s. The red lines represents the x axis of the sensor, the blue solid lines the y axis and the yellow lines the z axis.



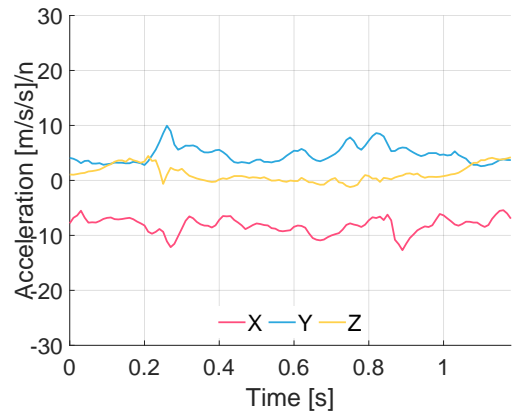
(a) Walking



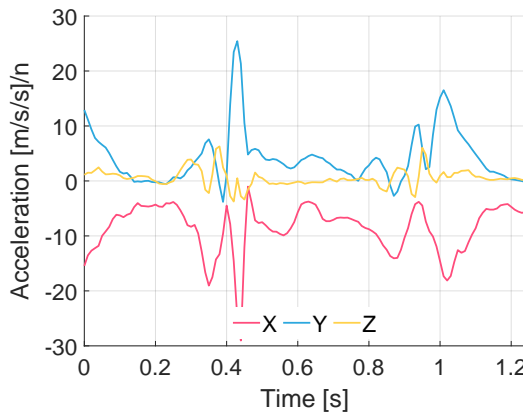
(b) Stopped



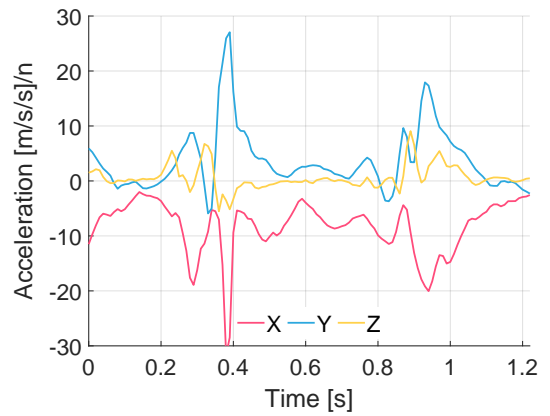
(c) Stair Ascent



(d) Stair Descent

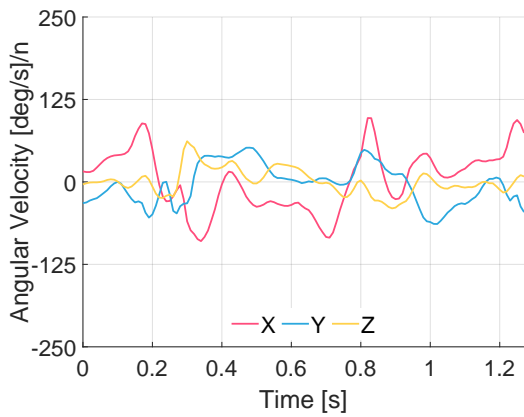


(e) Ramp Ascent

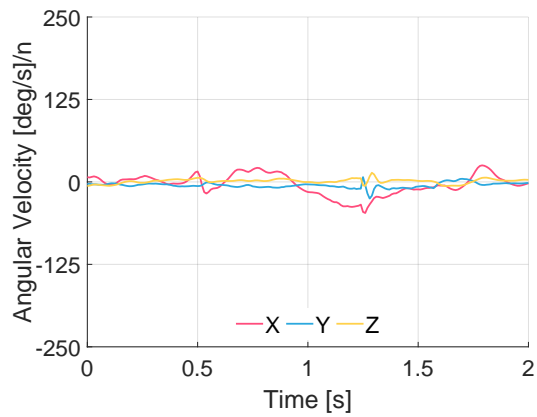


(f) Ramp Descent

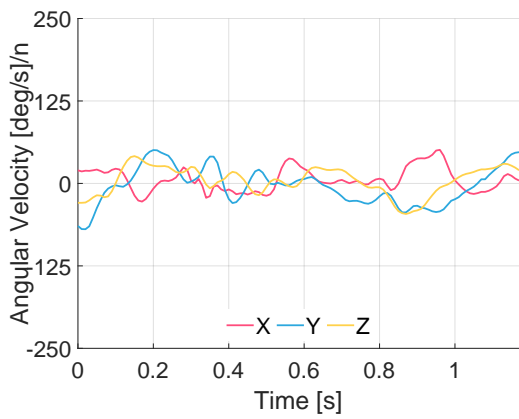
Figure D-5: Example data for the left hip accelerometer. The x represent recording time in seconds. The y axis show the measured acceleration in $m/s/s$. The red lines represents the x axis of the sensor, the blue solid lines the y axis and the yellow lines the z axis.



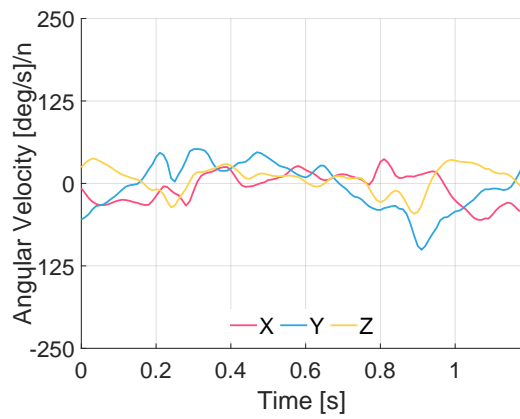
(a) Walking



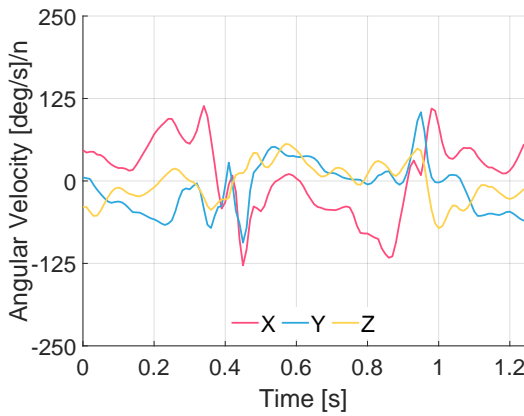
(b) Stopped



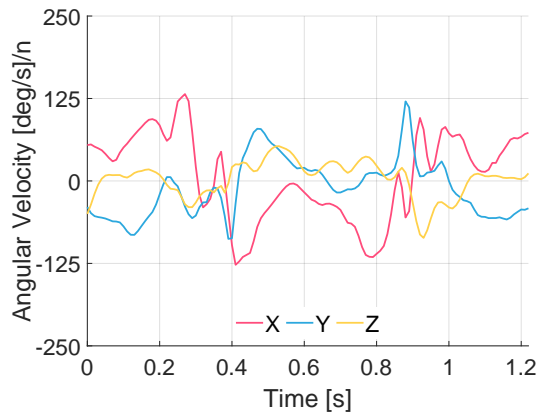
(c) Stair Ascent



(d) Stair Descent

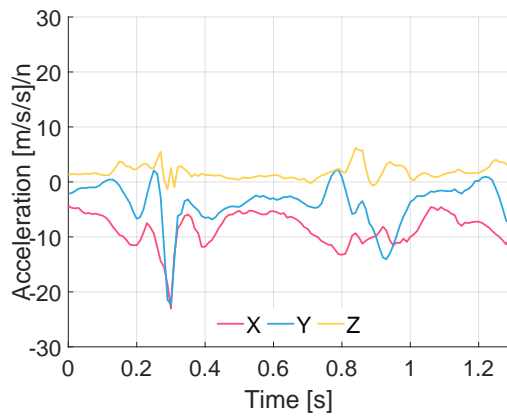


(e) Ramp Ascent

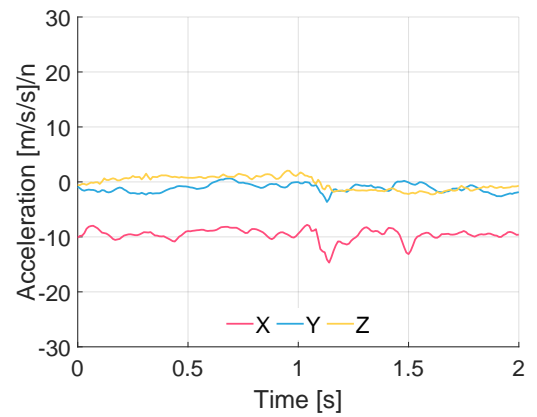


(f) Ramp Descent

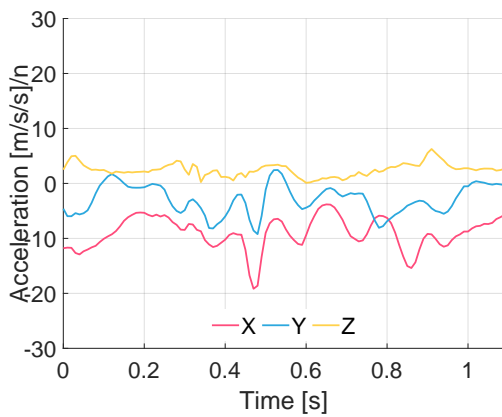
Figure D-6: Example data for the left hip gyroscope. The x represent recording time in seconds. The y axis show the measured angular velocity in deg/s. The red lines represents the x axis of the sensor, the blue solid lines the y axis and the yellow lines the z axis.



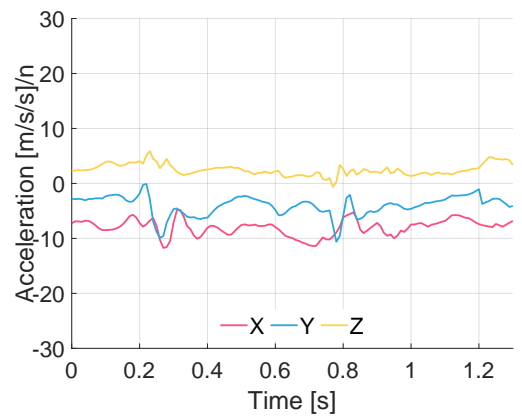
(a) Walking



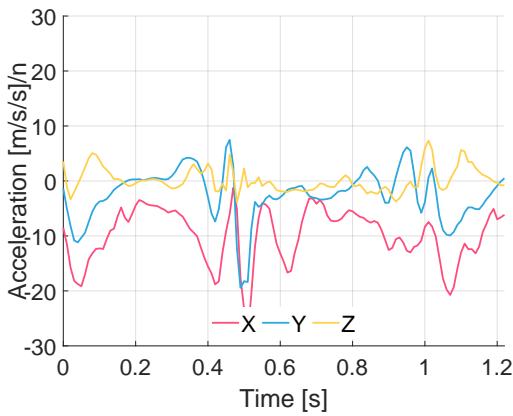
(b) Stopped



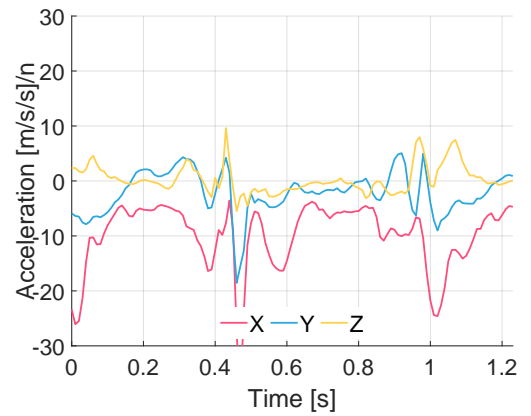
(c) Stair Ascent



(d) Stair Descent

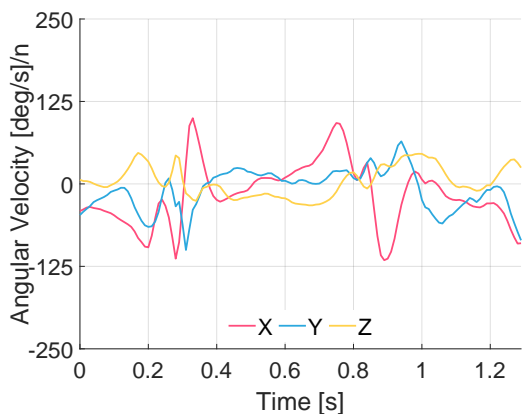


(e) Ramp Ascent

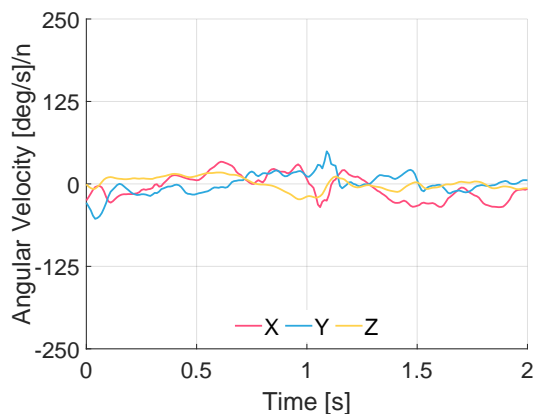


(f) Ramp Descent

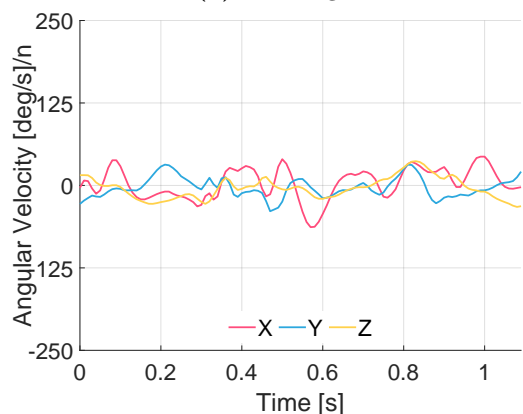
Figure D-7: Example data for the right hip accelerometer. The x represent recording time in seconds. The y axis show the measured acceleration in $m/s/s/n$. The red lines represents the x axis of the sensor, the blue solid lines the y axis and the yellow lines the z axis.



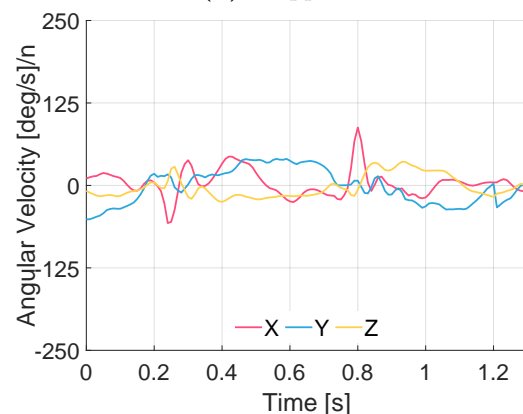
(a) Walking



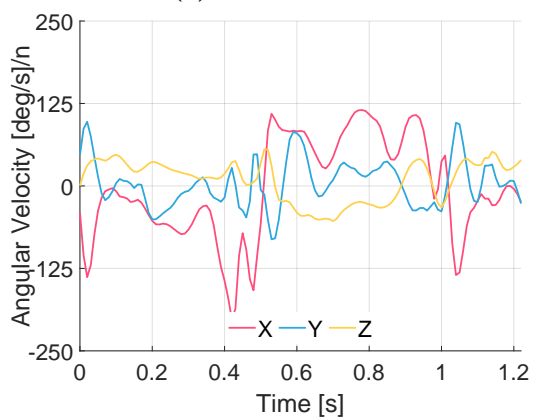
(b) Stopped



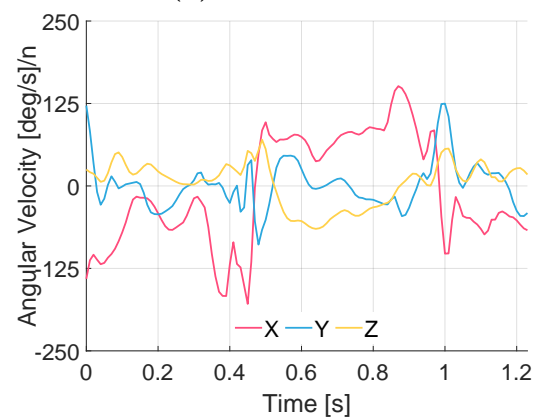
(c) Stair Ascent



(d) Stair Descent

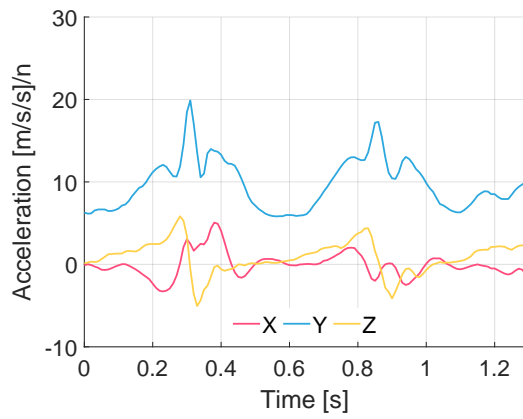


(e) Ramp Ascent

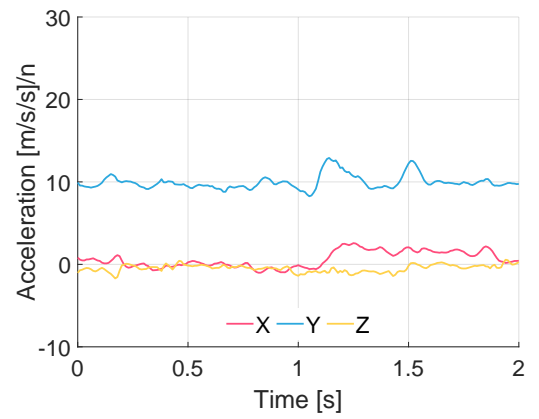


(f) Ramp Descent

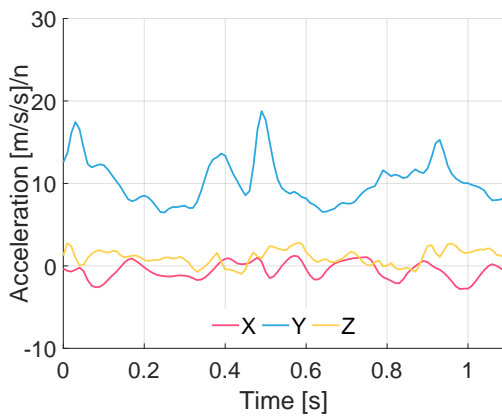
Figure D-8: Example data for the right hip gyroscope. The x represent recording time in seconds. The y axis show the measured angular velocity in deg/s. The red lines represents the x axis of the sensor, the blue solid lines the y axis and the yellow lines the z axis.



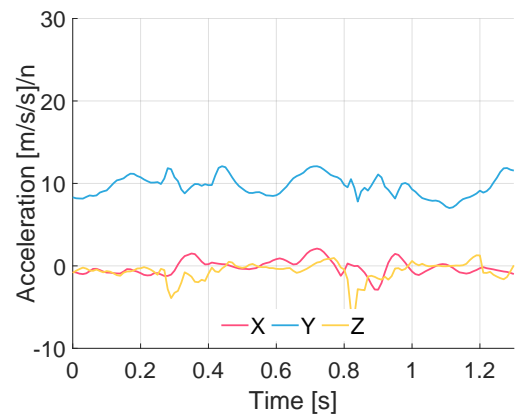
(a) Walking



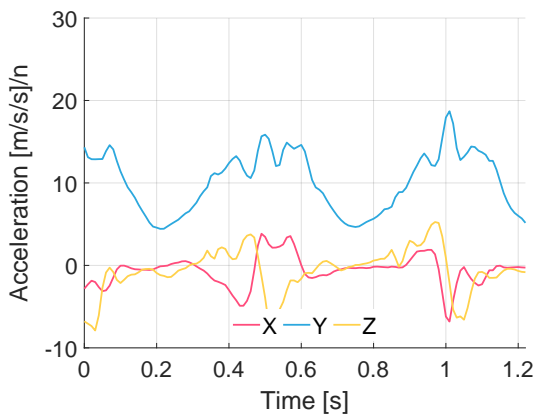
(b) Stopped



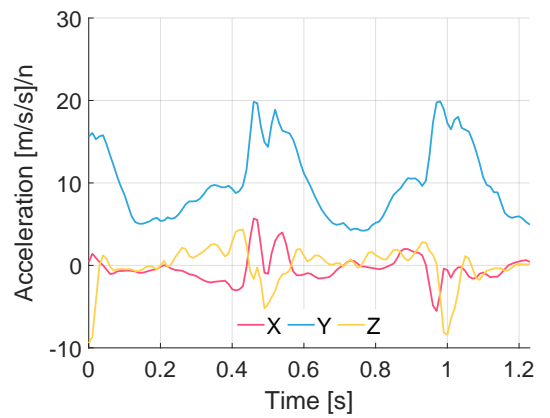
(c) Stair Ascent



(d) Stair Descent

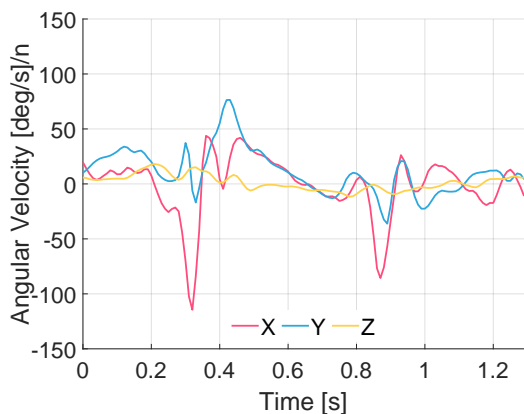


(e) Ramp Ascent

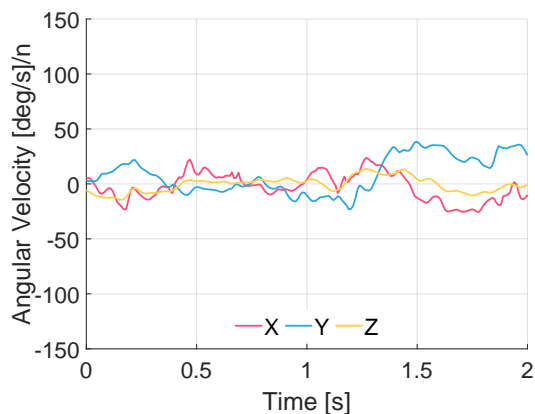


(f) Ramp Descent

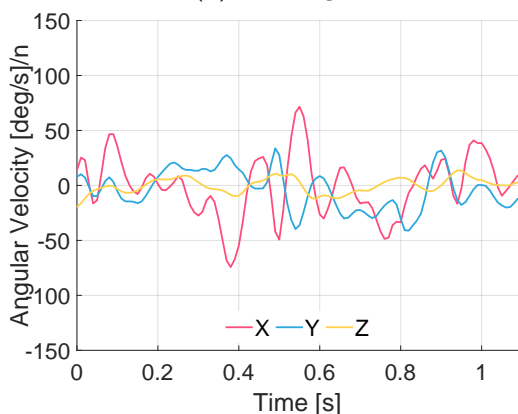
Figure D-9: Example data for the chest accelerometer. The x represent recording time in seconds. The y axis show the measured acceleration in $m/s/s$. The red lines represents the x axis of the sensor, the blue solid lines the y axis and the yellow lines the z axis.



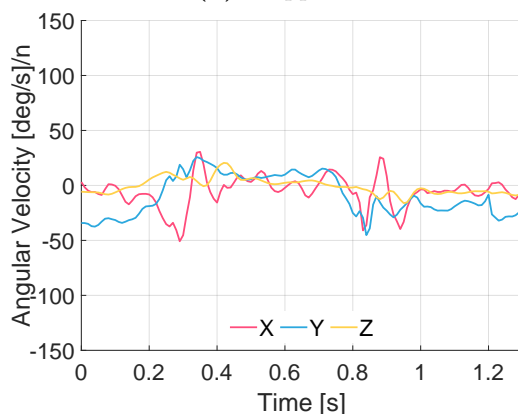
(a) Walking



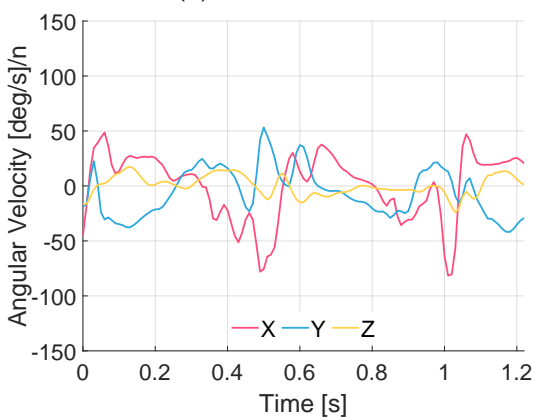
(b) Stopped



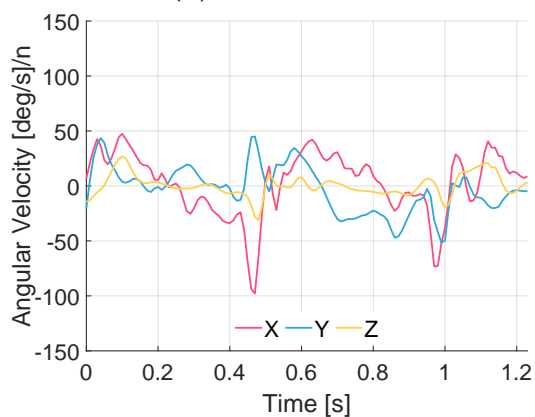
(c) Stair Ascent



(d) Stair Descent



(e) Ramp Ascent



(f) Ramp Descent

Figure D-10: Example data for the chest gyroscope. The x represent recording time in seconds. The y axis show the measured angular velocity in deg/s. The red lines represents the x axis of the sensor, the blue solid lines the y axis and the yellow lines the z axis.

References

- [1] Muhammad Asif et al. “Advancements, Trends and Future Prospects of Lower Limb Prosthesis”. In: *IEEE Access* 9 (2021). ISSN: 21693536. DOI: 10.1109/ACCESS.2021.3086807.
- [2] Kathryn J Griffin et al. “Toe Amputation: A predictor of future limb loss?” In: *Journal of Diabetes and its Complications* 26.3 (2012). ISSN: 10568727. DOI: 10.1016/j.jdiacomp.2012.03.003.
- [3] Nike Walter, Volker Alt, and Markus Rupp. “Lower Limb Amputation Rates in Germany”. In: *Medicina* 58.1 (Jan. 2022). ISSN: 1648-9144. DOI: 10.3390/medicina58010101.
- [4] Teuta Osmani Vllasolli et al. “Energy expenditure and walking speed in lower limb amputees: A cross sectional study”. In: *Ortopedia Traumatologia Rehabilitacja* 16.4 (2014). ISSN: 15093492. DOI: 10.5604/15093492.1119619.
- [5] Cody L. McDonald et al. “Energy expenditure in people with transtibial amputation walking with crossover and energy storing prosthetic feet: A randomized within-subject study”. In: *Gait and Posture* 62.March (2018). ISSN: 18792219. DOI: 10.1016/j.gaitpost.2018.03.040.
- [6] Hugh M. Herr and Alena M. Grabowski. “Bionic ankle-foot prosthesis normalizes walking gait for persons with leg amputation”. In: *Proceedings of the Royal Society B: Biological Sciences* 279.1728 (2012). ISSN: 14712970. DOI: 10.1098/rspb.2011.1194.
- [7] C Stewart. “Synopsis of Causation Lower Limb Amputation”. In: *Ministry Of Defence* September 2008 (2008).
- [8] Robert D. Gregg et al. “Virtual constraint control of a powered prosthetic leg: From simulation to experiments with transfemoral amputees”. In: *IEEE Transactions on Robotics* 30.6 (2014). ISSN: 15523098. DOI: 10.1109/TR0.2014.2361937.

- [9] Christopher K. Wong et al. “A Scoping Review of Physical Activity in People With Lower-Limb Loss: 10,000 Steps Per Day?” In: *Physical therapy* 101.8 (2021). ISSN: 15386724. DOI: 10.1093/ptj/pzab115.
- [10] Bantoon Srisuwan and Glenn K. Klute. “Locomotor activities of individuals with lower-limb amputation”. In: *Prosthetics and orthotics international* 45.3 (2021). ISSN: 17461553. DOI: 10.1097/PXR.000000000000009.
- [11] Suh Jen Lin et al. “Physical activity, functional capacity, and step variability during walking in people with lower-limb amputation”. In: *Gait and Posture* 40.1 (2014). ISSN: 18792219. DOI: 10.1016/j.gaitpost.2014.03.012.
- [12] Michael R. Tucker et al. “Control strategies for active lower extremity prosthetics and orthotics: A review”. In: *Journal of NeuroEngineering and Rehabilitation* 12.1 (2015). ISSN: 17430003. DOI: 10.1186/1743-0003-12-1.
- [13] Xin Zeng et al. “Gait-Based Implicit Authentication Using Edge Computing and Deep Learning for Mobile Devices”. In: *Sensors (Basel, Switzerland)* 21.13 (2021). ISSN: 14248220. DOI: 10.3390/s21134592.
- [14] Jaerock Kwon, Yunju Lee, and Jehyung Lee. “Comparative study of markerless vision-based gait analyses for person re-identification”. In: *Sensors* 21.24 (2021). ISSN: 14248220. DOI: 10.3390/s21248208.
- [15] Robert L. Waters and Sara Mulroy. “The energy expenditure of normal and pathologic gait”. In: *Gait and Posture* 9 (1999). ISSN: 09666362. DOI: 10.1016/S0966-6362(99)00009-0.
- [16] Fabian Horst et al. “Explaining the unique nature of individual gait patterns with deep learning”. In: *Scientific Reports* 9.1 (2019). ISSN: 20452322. DOI: 10.1038/s41598-019-38748-8.
- [17] Kanishk Shah, Matthew Solan, and Edward Dawe. “The gait cycle and its variations with disease and injury”. In: *Orthopaedics and Trauma* 34.3 (2020). ISSN: 18771335. DOI: 10.1016/j.mporth.2020.03.009.
- [18] Carlotta Mummolo, Luigi Mangialardi, and Joo H. Kim. “Quantifying dynamic characteristics of human walking for comprehensive gait cycle”. In: *Journal of Biomechanical Engineering* 135.9 (2013). ISSN: 01480731. DOI: 10.1115/1.4024755.
- [19] Levi J. Hargrove et al. “Intuitive control of a powered prosthetic leg during ambulation: A randomized clinical trial”. In: *JAMA - Journal of the American Medical Association* 313 (2015). ISSN: 15383598. DOI: 10.1001/jama.2015.4527.

- [20] Levi J. Hargrove et al. “Robotic Leg Control with EMG Decoding in an Amputee with Nerve Transfers”. In: *New England Journal of Medicine* 369 (2013). DOI: 10.1056/NEJMoa1300126.
- [21] Aaron J. Young et al. “Classifying the intent of novel users during human locomotion using powered lower limb prostheses”. In: *International IEEE/EMBS Conference on Neural Engineering, NER* (2013), pp. 311–314. DOI: 10.1109/NER.2013.6695934.
- [22] Ann M. Simon et al. “Strategies to reduce the configuration time for a powered knee and ankle prosthesis across multiple ambulation modes”. In: *IEEE International Conference on Rehabilitation Robotics* (2013). ISSN: 19457898. DOI: 10.1109/ICORR.2013.6650371.
- [23] Anoop Kant Godiyal et al. “Force myography based novel strategy for locomotion classification”. In: *IEEE Transactions on Human-Machine Systems* 48 (2018). ISSN: 21682291. DOI: 10.1109/THMS.2018.2860598.
- [24] Vijeth Rai, Abhishek Sharma, and Eric Rombokas. “Mode-free Control of Prosthetic Lower Limbs”. In: *2019 International Symposium on Medical Robotics, ISMR 2019* (2019). DOI: 10.1109/ISMR.2019.8710187.
- [25] Ben Yue Su et al. “A cnn-based method for intent recognition using inertial measurement units and intelligent lower limb prosthesis”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 27.5 (2019). ISSN: 15580210. DOI: 10.1109/TNSRE.2019.2909585.
- [26] Floriant Labarrière et al. “Machine Learning Approaches for Activity Recognition and/or Activity Prediction in Locomotion Assistive Devices—A Systematic Review”. In: *Sensors* 20.21 (Nov. 2020). ISSN: 1424-8220. DOI: 10.3390/s20216345.
- [27] Julian Bruinsma and Raffaella Carloni. “IMU-Based Deep Neural Networks: Prediction of Locomotor and Transition Intentions of an Osseointegrated Transfemoral Amputee”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 29 (2021). ISSN: 15580210. DOI: 10.1109/TNSRE.2021.3086843.
- [28] Markus Ebner et al. “Recognition of typical locomotion activities based on the sensor data of a smartphone in pocket or hand”. In: *Sensors (Switzerland)* 20 (2020). ISSN: 14248220. DOI: 10.3390/s20226559.
- [29] Tom F Novacheck. “The biomechanics of running”. In: *Gait & Posture* 7.1 (Jan. 1998). ISSN: 09666362. DOI: 10.1016/S0966-6362(97)00038-6.
- [30] Freddie Sherratt, Andrew Plummer, and Pejman Irvani. “Understanding LSTM Network Behaviour of IMU-Based”. In: *Sensors* 21.1264 (2021). DOI: 10.3390/s21041264.

- [31] Roger Bartlett. *Introduction to Sports Biomechanics: Analysing Human Movement Patterns*. 2nd. Vol. 50 Suppl 1. Routledge, 2007.
- [32] Michael Whittle, David Levine, and Jim Richards. *Whittle's Gait analysis*. eng. 5th ed. Edinburgh: Churchill Livingstone Elsevier, 2012. ISBN: 9780702042652.
- [33] Tyagi Ramakrishnan et al. "Human Gait Analysis Metric for Gait Retraining". In: *Applied Bionics and Biomechanics* 2019 (2019). ISSN: 17542103. DOI: 10.1155/2019/1286864.
- [34] Fiona Coutts. "Gait analysis in the therapeutic environment". In: *Manual Therapy* 4.1 (Feb. 1999). ISSN: 1356689X. DOI: 10.1016/S1356-689X(99)80003-4.
- [35] Jason R. Franz and Rodger Kram. "The effects of grade and speed on leg muscle activations during walking". In: *Gait and Posture* 35.1 (2012). ISSN: 09666362. DOI: 10.1016/j.gaitpost.2011.08.025.
- [36] Wolfgang Svensson and Ulf Holmberg. "Stair Gait Classification from Kinematic Sensors". In: *Rehabilitation Robotics* (2007). DOI: 10.5772/5178.
- [37] Ada S. Cheung et al. "Biomechanical Leg Muscle Function during Stair Ambulation in Men Receiving Androgen Deprivation Therapy". In: *Journals of Gerontology - Series A Biological Sciences and Medical Sciences* 75 (2020). ISSN: 1758535X. DOI: 10.1093/gerona/glz169.
- [38] Thomas C. Bulea et al. "Forward stair descent with hybrid neuroprosthesis after paralysis: Single case study demonstrating feasibility". In: *Journal of Rehabilitation Research and Development* 51 (2014). ISSN: 19381352. DOI: 10.1682/JRRD.2013.12.0257.
- [39] Keigo Matsumoto et al. "Walking uphill and downhill: Redirected walking in the vertical direction". In: *ACM SIGGRAPH 2017 Posters, SIGGRAPH 2017* (2017). DOI: 10.1145/3102163.3102227.
- [40] Shani Kimel-Naor, Amihai Gottlieb, and Meir Plotnik. "The effect of uphill and downhill walking on gait parameters: A self-paced treadmill study". In: *Journal of Biomechanics* 60 (2017). ISSN: 18732380. DOI: 10.1016/j.jbiomech.2017.06.030.
- [41] Andrew Stuart McIntosh et al. "Gait dynamics on an inclined walkway". In: *Journal of Biomechanics* 39.13 (2006). ISSN: 00219290. DOI: 10.1016/j.jbiomech.2005.07.025.
- [42] Jun Huai Li et al. "Segmentation and Recognition of Basic and Transitional Activities for Continuous Physical Human Activity". In: *IEEE Access* 7 (2019). ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2905575.

- [43] Marco Rabuffetti, M. Recalcati, and M. Ferrarin. “Trans-femoral amputee gait: Socket-pelvis constraints and compensation strategies”. In: *Prosthetics and Orthotics International* 29 (Aug. 2005). ISSN: 03093646. DOI: 10.1080/03093640500217182.
- [44] Anne K. Silverman et al. “Compensatory mechanisms in below-knee amputee gait in response to increasing steady-state walking speeds”. In: *Gait and Posture* 28.4 (2008). ISSN: 09666362. DOI: 10.1016/j.gaitpost.2008.04.005.
- [45] K. Lechler and K. Kristjansson. “The importance of additional mid swing toe clearance for amputees”. In: *Canadian Prosthetics and Orthotics Journal* 1 (2018). ISSN: 2561987X. DOI: 10.33137/cpoj.v1i2.30813.
- [46] Ida Kovac et al. “Ground reaction force analysis in traumatic transtibial amputees’ gait”. In: *Collegium antropologicum* 33 Suppl 2 (2009).
- [47] Brendan Burkett, James Smeathers, and Timothy M. Barker. “A Computer Model to Simulate the Swing Phase of a Transfemoral Prosthesis”. In: *Journal of Applied Biomechanics* 20 (2004). DOI: 10.1123/jab.20.1.25.
- [48] Hamid Bateni and Sandra J. Olney. “Kinematic and kinetic variations of below-knee amputee gait”. In: *Journal of Prosthetics and Orthotics* 14.1 (2002). ISSN: 10408800. DOI: 10.1097/00008526-200203000-00003.
- [49] Tiwana Varrecchia et al. “Common and specific gait patterns in people with varying anatomical levels of lower limb amputation and different prosthetic components”. In: *Human Movement Science* 66.October 2018 (2019). ISSN: 18727646. DOI: 10.1016/j.humov.2019.03.008.
- [50] Peter Gabriel Adamczyk and Arthur D. Kuo. “Mechanisms of gait asymmetry due to push-off deficiency in unilateral amputees”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 23.5 (2015). ISSN: 15344320. DOI: 10.1109/TNSRE.2014.2356722.
- [51] Vahidreza Jafari Harandi et al. “Gait compensatory mechanisms in unilateral transfemoral amputees”. In: *Medical Engineering and Physics* 77 (2020). ISSN: 18734030. DOI: 10.1016/j.medengphy.2019.11.006.
- [52] Laura Hak et al. “Stepping asymmetry among individuals with unilateral transtibial limb loss might be functional in terms of gait stability”. In: *Physical Therapy* 94.10 (2014). ISSN: 15386724. DOI: 10.2522/ptj.20130431.
- [53] Johannes B Bussmann, Hannelore J Schrauwen, and Henk J Stam. “Daily physical activity and heart rate response in people with a unilateral traumatic transtibial amputation”. In: *Archives of Physical Medicine and Rehabilitation* 89.3 (2008). ISSN: 1532-821X; 0003-9993. DOI: 10.1016/j.apmr.2007.11.012[doi].

- [54] L. Piazza et al. “Assesment of physical activity in amputees: A systematic review of the literature”. In: *Science & Sports* 32.4 (Sept. 2017). ISSN: 07651597. DOI: 10.1016/j.scispo.2017.07.011.
- [55] Samuel K Au, Jeff Weber, and Hugh Herr. “Powered ankle-foot prosthesis improves walking metabolic economy”. In: *IEEE Transactions on Robotics* 25.1 (2009). ISSN: 15523098. DOI: 10.1109/TR0.2008.2008747.
- [56] Chris Button, Stuart Moyle, and Keith Davids. “Comparison of below-knee amputee gait performed overground and on a motorized treadmill”. In: *Adapted Physical Activity Quarterly* 27.2 (2010). ISSN: 15432777. DOI: 10.1123/apaq.27.2.96.
- [57] Sonja M.H.J. Jaegers, J. Hans Arendzen, and Henry J. de Jongh. “Prosthetic gait of unilateral transfemoral amputees: A kinematic study”. In: *Archives of Physical Medicine and Rehabilitation* 76 (1995). ISSN: 00039993. DOI: 10.1016/S0003-9993(95)80528-1.
- [58] A M Boonstra, V Fidler, and W H Eisma. “Walking speed of normal subjects and amputees: Aspects of validity of gait analysis”. In: *Prosthetics and Orthotics International* 17.2 (1993). ISSN: 17461553. DOI: 10.3109/03093649309164360.
- [59] Xavier Drevelle et al. “Vaulting quantification during level walking of transfemoral amputees”. In: *Clinical Biomechanics* 29 (2014). ISSN: 18791271. DOI: 10.1016/j.clinbiomech.2014.04.006.
- [60] Lee Nolan et al. “Adjustments in gait symmetry with walking speed in trans-femoral and trans-tibial amputees”. In: *Gait and Posture* 17 (2003). ISSN: 09666362. DOI: 10.1016/S0966-6362(02)00066-8.
- [61] Claudia Ochoa-Diaz and Antônio Padilha L. Bó. “Symmetry analysis of amputee gait based on body center of mass trajectory and discrete fourier transform”. In: *Sensors (Switzerland)* 20.8 (2020). ISSN: 14248220. DOI: 10.3390/s20082392.
- [62] John Kirkup. *A History of Limb Amputation*. London: Springer London, 2007. ISBN: 978-1-84628-443-4. DOI: 10.1007/978-1-84628-509-7.
- [63] James Breakey. “Gait of Unilateral Below-Knee Amputees”. In: *Orthotics and Prosthetics* 30.3 (1976).
- [64] Blatchford. *Prosthetic Feet - Prosthetic Product Catalogue - Blatchford*. URL: <https://www.blatchford.co.uk/products/category/prosthetics/feet/> (Accessed on 01/28/2021).
- [65] Ottobock. *Empower — Ottobock UK*. URL: <https://www.ottobock.co.uk/prosthetics/lower-limb-prosthetics/prosthetic-product-systems/empower-ankle/> (Accessed on 07/31/2018).

- [66] Krishan Bhakta et al. “Machine Learning Model Comparisons of User Independent & Dependent Intent Recognition Systems for Powered Prostheses”. In: *IEEE Robotics and Automation Letters* 5.4 (2020). ISSN: 23773766. DOI: 10.1109/LRA.2020.3007480.
- [67] Smita Nayak and Rajesh Kumar Das. “Application of Artificial Intelligence (AI) in Prosthetic and Orthotic Rehabilitation”. In: *Service Robotics* (2020). DOI: 10.5772/intechopen.93903.
- [68] Össur. *Össur Iberia - Life without limitations*. URL: <https://www.ossur.co.uk/> (Accessed on 09/17/2018).
- [69] Andrea N. Lay et al. “The effects of sloped surfaces on locomotion: An electromyographic analysis”. In: *Journal of Biomechanics* 40.6 (2007). ISSN: 00219290. DOI: 10.1016/j.jbiomech.2006.05.023.
- [70] Ivan Hernandez and Wen Yu. “Recent Advances on Control of Active Lower Limb Prostheses”. In: *IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India)* (2021). ISSN: 09745971. DOI: 10.1080/02564602.2021.1994477.
- [71] Huseyin Atakan Varol, Frank Sup, and Michael Goldfarb. “Multiclass real-time intent recognition of a powered lower limb prosthesis”. In: *IEEE Transactions on Biomedical Engineering* 57 (2010). ISSN: 00189294. DOI: 10.1109/TBME.2009.2034734.
- [72] Mahdiah Kazemimoghadam and Nicholas P. Fey. “Continuous Classification of Locomotion in Response to Task Complexity and Anticipatory State”. In: *Frontiers in Bioengineering and Biotechnology* 9. April (2021). ISSN: 22964185. DOI: 10.3389/fbioe.2021.628050.
- [73] Kaitlin G. Rabe et al. “Ultrasound Sensing Can Improve Continuous Classification of Discrete Ambulation Modes Compared to Surface Electromyography”. In: *IEEE Transactions on Biomedical Engineering* 68 (2021). ISSN: 15582531. DOI: 10.1109/TBME.2020.3032077.
- [74] Minhan Li et al. “Fusion of Human Gaze and Machine Vision for Predicting Intended Locomotion Mode”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 30 (2022). ISSN: 15580210. DOI: 10.1109/TNSRE.2022.3168796.
- [75] Jindong Wang et al. “Deep learning for sensor-based activity recognition: A survey”. In: *Pattern Recognition Letters* 119 (2019). ISSN: 01678655. DOI: 10.1016/j.patrec.2018.02.010.

- [76] Vijeth Rai and Eric Rombokas. “A framework for mode-free prosthetic control for unstructured terrains”. In: *IEEE International Conference on Rehabilitation Robotics* 2019-June (2019). DOI: 10.1109/ICORR.2019.8779439.
- [77] Harrison L. Bartlett et al. “A Semi-Powered Ankle Prosthesis and Unified Controller for Level and Sloped Walking”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 29 (2021). ISSN: 15580210. DOI: 10.1109/TNSRE.2021.3049194.
- [78] Tian Yu et al. “The Design, Control, and Testing of an Integrated Electrohydrostatic Powered Ankle Prosthesis”. In: *IEEE/ASME Transactions on Mechatronics* 24.3 (2019). ISSN: 10834435. DOI: 10.1109/TMECH.2019.2911685.
- [79] Frank Sup, Amit Bohara, and Michael Goldfarb. “Design and control of a powered transfemoral prosthesis”. In: *International Journal of Robotics Research* 27 (2008). ISSN: 02783649. DOI: 10.1177/0278364907084588.
- [80] Fei Gao, Yannan Liu, and Wei Hsin Liao. “Implementation and testing of ankle-foot prosthesis with a new compensated controller”. In: *IEEE/ASME Transactions on Mechatronics* 24 (2020). ISSN: 1941014X.
- [81] Abdul Hadi Abdul Razak et al. “Foot plantar pressure measurement system: A review”. In: *Sensors (Switzerland)* 12.7 (2012). ISSN: 14248220. DOI: 10.3390/s120709884.
- [82] Jeffrey R Koller, C. David Remy, and Daniel P Ferris. “Biomechanics and energetics of walking in powered ankle exoskeletons using myoelectric control versus mechanically intrinsic control”. In: *Journal of NeuroEngineering and Rehabilitation* 15.1 (2018). ISSN: 17430003. DOI: 10.1186/s12984-018-0379-6.
- [83] Pete B Shull et al. “Quantified self and human movement: A review on the clinical impact of wearable sensing and feedback for gait analysis and intervention”. In: 40.1 (2014). ISSN: 18792219. DOI: 10.1016/j.gaitpost.2014.03.189.
- [84] H. Martin Schepers, H. F J M Koopman, and Peter H. Veltink. “Ambulatory assessment of ankle and foot dynamics”. In: *IEEE Transactions on Biomedical Engineering* 54.5 (2007). ISSN: 00189294. DOI: 10.1109/TBME.2006.889769.
- [85] Frieder Wittmann, Olivier Lambercy, and Roger Gassert. “Magnetometer-based drift correction during rest in IMU arm motion tracking”. In: *Sensors (Switzerland)* 19 (2019). ISSN: 14248220. DOI: 10.3390/s19061312.

- [86] Srikanth Sagar Bangaru, Chao Wang, and Fereydoun Aghazadeh. “Data quality and reliability assessment of wearable emg and IMU sensor for construction activity recognition”. In: *Sensors (Switzerland)* 20 (2020). ISSN: 14248220. DOI: 10.3390/s20185264.
- [87] Huaijun Wang et al. “Wearable Sensor-Based Human Activity Recognition Using Hybrid Deep Learning Techniques”. In: *Security and Communication Networks* 2020 (July 2020). ISSN: 19390122. DOI: 10.1155/2020/2132138.
- [88] Angelo Maria Sabatini and Vincenzo Genovese. “A sensor fusion method for tracking vertical velocity and height based on inertial and barometric altimeter measurements”. In: *Sensors (Switzerland)* 14 (2014). ISSN: 14248220. DOI: 10.3390/s140813324.
- [89] Odongo Steven Eyobu and Dong Seog Han. “Feature representation and data augmentation for human activity classification based on wearable IMU sensor data using a deep LSTM neural network”. In: *Sensors (Switzerland)* 18 (2018). ISSN: 14248220. DOI: 10.3390/s18092892.
- [90] Binbin Su and Elena M. Gutierrez-Farewik. “Gait trajectory and gait phase prediction based on an LSTM network”. In: *Sensors (Switzerland)* 20 (2020). ISSN: 14248220. DOI: 10.3390/s20247127.
- [91] Francisco Javier Ordóñez and Daniel Roggen. “Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition”. In: *Sensors (Switzerland)* 16.1 (2016). ISSN: 14248220. DOI: 10.3390/s16010115.
- [92] Roy R. Minor and David W. Rowe. “Utilization of GPS/MEMS-IMU for measurement of dynamics for range testing of missiles and rockets”. In: *Record - IEEE PLANS, Position Location and Navigation Symposium* (1998). DOI: 10.1109/plans.1998.670219.
- [93] W. Geiger et al. “MEMS IMU for AHRS applications”. In: *Record - IEEE PLANS, Position Location and Navigation Symposium* (2008). DOI: 10.1109/PLANS.2008.4569973.
- [94] Digikey. *ADXL356BEZ-RL7 Analog Devices Inc.* — *Sensors, Transducers* — *DigiKey*. URL: <https://www.digikey.co.uk/en/products/detail/analog-devices-inc./ADXL356BEZ-RL7/13544950> (Accessed on 08/11/2022).
- [95] Slavomir Kardos, Peter Balog, and Stanislav Slosarcik. “Gait dynamics sensing using IMU sensor array system”. In: *Advances in Electrical and Electronic Engineering* 15 (2017). ISSN: 18043119. DOI: 10.15598/aeec.v15i1.2019.

- [96] Paola Catalfamo Formento et al. “Gait event detection during stair walking using a rate gyroscope”. In: *Sensors (Switzerland)* 14 (2014). ISSN: 14248220. DOI: 10.3390/s140305470.
- [97] X. Jiang et al. “Exploration of Gait Parameters Affecting the Accuracy of Force Myography-Based Gait Phase Detection”. In: *Proceedings of the IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechanics 2018-Augus* (2018). ISSN: 21551774. DOI: 10.1109/BIOROB.2018.8487790.
- [98] Anoop Kant Godiyal et al. “A force myography-based system for gait event detection in overground and ramp walking”. In: *IEEE Transactions on Instrumentation and Measurement* 67 (2018). ISSN: 00189456. DOI: 10.1109/TIM.2018.2816799.
- [99] Yongtian He et al. “Brain-machine interfaces for controlling lower-limb powered robotic systems”. In: 15.2 (2018). ISSN: 17412552. DOI: 10.1088/1741-2552/aaa8c0.
- [100] He Huang et al. “Continuous locomotion-mode identification for prosthetic legs based on neuromuscular - Mechanical fusion”. In: *IEEE Transactions on Biomedical Engineering* 58 (2011). ISSN: 00189294. DOI: 10.1109/TBME.2011.2161671.
- [101] A. Rainoldi, G. Melchiorri, and I. Caruso. “A method for positioning electrodes during surface EMG recordings in lower limb muscles”. In: *Journal of Neuroscience Methods* 134 (2004). ISSN: 01650270. DOI: 10.1016/j.jneumeth.2003.10.014.
- [102] *SENIAM. Project - Sensor placement*. 2005. URL: <http://www.seniam.org/> (Accessed on 08/11/2022).
- [103] Toby Elery et al. “Design and validation of a powered knee-ankle prosthesis with high-torque, low-impedance actuators”. In: *IEEE Transactions on Robotics* 36 (2020). ISSN: 19410468. DOI: 10.1109/TR0.2020.3005533.
- [104] Baojun Chen, Yanggang Feng, and Qining Wang. “Combining vibrotactile feedback with volitional myoelectric control for robotic transtibial prostheses”. In: *Frontiers in Neurorobotics* 10.8 (2016). ISSN: 16625218. DOI: 10.3389/fnbot.2016.00008.
- [105] Michael Tschiedel, Michael Friedrich Russold, and Eugenijus Kaniusas. “Relying on more sense for enhancing lower limb prostheses control: a review”. In: *Journal of neuroengineering and rehabilitation* 17.1 (2020). ISSN: 17430003. DOI: 10.1186/s12984-020-00726-x.
- [106] Ming Jin et al. “Environmental sensing by wearable device for indoor activity and location estimation”. In: *IECON Proceedings (Industrial Electronics Conference)* (2014). ISSN: 1553-572X. DOI: 10.1109/IECON.2014.7049320.

- [107] Zhongzheng Fu et al. “Personalized human activity recognition based on integrated wearable sensor and transfer learning”. In: *Sensors (Switzerland)* 21.3 (2021). ISSN: 14248220. DOI: 10.3390/s21030885.
- [108] Texas Instruments. *TMP117 data sheet, product information and support — TI.com*. URL: <https://www.ti.com/product/TMP117> (Accessed on 08/11/2022).
- [109] Seungeun Chung et al. “Sensor data acquisition and multimodal sensor fusion for human activity recognition using deep learning”. In: *Sensors (Switzerland)* 19.7 (2019). ISSN: 14248220. DOI: 10.3390/s19071716.
- [110] Yi Xing Liu, Ruoli Wang, and Elena M. Gutierrez-Farewik. “A Muscle Synergy-Inspired Method of Detecting Human Movement Intentions Based on Wearable Sensor Fusion”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 29 (2021). ISSN: 15580210. DOI: 10.1109/TNSRE.2021.3087135.
- [111] Fo Hu et al. “A novel fusion strategy for locomotion activity recognition based on multimodal signals”. In: *Biomedical Signal Processing and Control* 67. September 2020 (2021). ISSN: 17468108. DOI: 10.1016/j.bspc.2021.102524.
- [112] Xugang Xi et al. “Surface Electromyography-Based Daily Activity Recognition Using Wavelet Coherence Coefficient and Support Vector Machine”. In: *Neural Processing Letters* 50 (2019). ISSN: 1573773X. DOI: 10.1007/s11063-019-10008-w.
- [113] Ying Li et al. “Gait recognition based on EMG with different individuals and sample sizes”. In: *Chinese Control Conference, CCC 2016-Augus* (2016). ISSN: 21612927. DOI: 10.1109/ChiCC.2016.7553988.
- [114] Junyao Wang et al. “Research on Gait Recognition Based on Lower Limb EMG Signal”. In: *2021 IEEE International Conference on Mechatronics and Automation, ICMA 2021* (2021). DOI: 10.1109/ICMA52036.2021.9512759.
- [115] Dongbin Shin, Seungchan Lee, and Seunghoon Hwang. “Locomotion mode recognition algorithm based on gaussian mixture model using imu sensors”. In: *Sensors* 21.8 (2021). ISSN: 14248220. DOI: 10.3390/s21082785.
- [116] Haoyu Li, Stéphane Derrode, and Wojciech Pieczynski. “An adaptive and on-line IMU-based locomotion activity classification method using a triplet Markov model”. In: *Neurocomputing* 362 (2019). ISSN: 18728286. DOI: 10.1016/j.neucom.2019.06.081.
- [117] Yang Han et al. “Design of decision tree structure with improved BPNN nodes for high-accuracy locomotion mode recognition using a single IMU”. In: *Sensors (Switzerland)* 21 (2021). ISSN: 14248220. DOI: 10.3390/s21020526.
- [118] Thomas Mitchell. *Machine learning*. McGraw-Hill, 1997. ISBN: 0070428077.

- [119] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge: Cambridge University Press, 2014. ISBN: 9781107298019. DOI: 10.1017/CB09781107298019.
- [120] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press, 2016.
- [121] Andriy Burkov. “The hundred-page machine learning book”. In: *Expert Systems* (May 2019). ISSN: 9781999579517.
- [122] Yoshua Bengio, Aaron Courville, and Pascal Vincent. “Representation learning: A review and new perspectives”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013). ISSN: 01628828. DOI: 10.1109/TPAMI.2013.50.
- [123] Zahraa S Abdallah and Mohamed Medhat Gaber. “Activity Recognition with Evolving Data Streams: A Review”. In: *ACM Comput. Surv* 51.4 (2018). ISSN: 03600300. DOI: 10.1145/3158645.
- [124] Abolfazl Farahani et al. “A Concise Review of Transfer Learning”. In: *Proceedings - 2020 International Conference on Computational Science and Computational Intelligence, CSCI 2020* (2020). DOI: 10.1109/CSCI51800.2020.00065.
- [125] Fuzhen Zhuang et al. “A Comprehensive Survey on Transfer Learning”. In: *Proceedings of the IEEE* 109.1 (2021). ISSN: 15582256. DOI: 10.1109/JPROC.2020.3004555.
- [126] Alex Sherstinsky. “Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network”. In: *Physica D: Nonlinear Phenomena* 404 (2020). ISSN: 01672789. DOI: 10.1016/j.physd.2019.132306.
- [127] Jinwon Lee, Woolim Hong, and Pilwon Hur. “Continuous Gait Phase Estimation Using LSTM for Robotic Transfemoral Prosthesis across Walking Speeds”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 29 (2021). ISSN: 15580210. DOI: 10.1109/TNSRE.2021.3098689.
- [128] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (1997). ISSN: 08997667. DOI: 10.1162/neco.1997.9.8.1735.
- [129] Klaus Greff et al. “LSTM: A Search Space Odyssey”. In: *IEEE Transactions on Neural Networks and Learning Systems* 28 (2017). ISSN: 21622388. DOI: 10.1109/TNNLS.2016.2582924.
- [130] Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. “Learning to Forget: Continual Prediction with LSTM”. In: *Neural Computation* 12 (Oct. 2000). ISSN: 0899-7667. DOI: 10.1162/089976600300015015.

- [131] Ran Zhu et al. “Efficient Human Activity Recognition Solving the Confusing Activities Via Deep Ensemble Learning”. In: *IEEE Access* 7 (2019). ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2922104.
- [132] Huseyin Atakan Varol, Frank Sup, and Michael Goldfarb. “Real-time gait mode intent recognition of a powered knee and ankle prosthesis for standing and walking”. In: *Proceedings of the 2nd Biennial IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, BioRob 2008*. 2008. ISBN: 9781424428830. DOI: 10.1109/BIOROB.2008.4762860.
- [133] Brian E. Lawson et al. “A robotic leg prosthesis: Design, control, and implementation”. In: *IEEE Robotics and Automation Magazine* 21.4 (2014). ISSN: 10709932. DOI: 10.1109/MRA.2014.2360303.
- [134] Maja Goršič et al. “Online phase detection using wearable sensors for walking with a robotic prosthesis”. In: *Sensors (Switzerland)* 14.2 (2014). ISSN: 14248220. DOI: 10.3390/s140202776.
- [135] Aaron J. Young, Ann M. Simon, and Levi J. Hargrove. “A training method for locomotion mode prediction using powered lower limb prostheses”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 22.3 (2014). ISSN: 15344320. DOI: 10.1109/TNSRE.2013.2285101.
- [136] Brian Coley et al. “Stair climbing detection during daily physical activity using a miniature gyroscope”. In: *Gait & Posture* 22.4 (2005). ISSN: 09666362. DOI: 10.1016/j.gaitpost.2004.08.008.
- [137] Chris Edwards. “Deep learning hunts for signals among the noise”. In: *Communications of the ACM* 61 (May 2018). ISSN: 0001-0782. DOI: 10.1145/3204445.
- [138] Binh Nguyen et al. “Trends in human activity recognition with focus on machine learning and power requirements”. In: *Machine Learning with Applications* 5.6 (2021). ISSN: 26668270.
- [139] Sen Qiu et al. “Multi-sensor information fusion based on machine learning for real applications in human activity recognition: State-of-the-art and research challenges”. In: *Information Fusion* 80.November 2021 (2022). ISSN: 15662535. DOI: 10.1016/j.inffus.2021.11.006.
- [140] Rashmi Saini and Vinod Maan. “Human Activity and Gesture Recognition: A Review”. In: *Proceedings - 2020 International Conference on Emerging Trends in Communication, Control and Computing, ICONC3 2020* (2020). DOI: 10.1109/ICONC345789.2020.9117535.

- [141] Marcin Straczekiewicz, Peter James, and Jukka Pekka Onnela. “A systematic review of smartphone-based human activity recognition methods for health research”. In: *npj Digital Medicine* 4.1 (2021). ISSN: 23986352. DOI: 10.1038/s41746-021-00514-4.
- [142] Patrícia Bota et al. “A semi-automatic annotation approach for human activity recognition”. In: *Sensors (Switzerland)* 19.3 (2019). ISSN: 14248220. DOI: 10.3390/s19030501.
- [143] Wenchao Jiang and Zhaozheng Yin. “Human activity recognition using wearable sensors by deep convolutional neural networks”. In: *MM 2015 - Proceedings of the 2015 ACM Multimedia Conference* (2015). DOI: 10.1145/2733373.2806333.
- [144] Huaitian Lu, Lambert R.B. Schomaker, and Raffaella Carloni. “IMU-based deep neural networks for locomotor intention prediction”. In: *IEEE International Conference on Intelligent Robots and Systems* (2020). ISSN: 21530866. DOI: 10.1109/IR0S45743.2020.9341649.
- [145] Uriel Martinez-Hernandez, Mohammed I. Awad, and Abbas A. Dehghani-Sanij. “Learning architecture for the recognition of walking and prediction of gait period using wearable sensors”. In: *Neurocomputing* (2021). ISSN: 09252312. DOI: 10.1016/j.neucom.2021.10.044.
- [146] Tao Yu et al. “A Multi-Layer Parallel LSTM Network for Human Activity Recognition with Smartphone Sensors”. In: *2018 10th International Conference on Wireless Communications and Signal Processing, WCSP 2018* (2018). DOI: 10.1109/WCSP.2018.8555945.
- [147] Md Zia Uddin and Ahmet Soylu. “Human activity recognition using wearable sensors, discriminant analysis, and long short-term memory-based neural structured learning”. In: *Scientific Reports* 11.1 (2021). ISSN: 20452322. DOI: 10.1038/s41598-021-95947-y.
- [148] Ronald Mutegeki and Dong Seog Han. “A CNN-LSTM Approach to Human Activity Recognition”. In: *2020 International Conference on Artificial Intelligence in Information and Communication, ICAIIC 2020* (2020). DOI: 10.1109/ICAIIC48513.2020.9065078.
- [149] Kun Xia, Jianguang Huang, and Hanyu Wang. “LSTM-CNN Architecture for Human Activity Recognition”. In: *IEEE Access* 8 (2020). ISSN: 21693536. DOI: 10.1109/ACCESS.2020.2982225.

- [150] Abdulmajid Murad and Jae Young Pyun. “Deep recurrent neural networks for human activity recognition”. In: *Sensors (Switzerland)* 17.11 (2017). ISSN: 14248220. DOI: 10.3390/s17112556.
- [151] Nilay Tufek et al. “Human Action Recognition Using Deep Learning Methods on Limited Sensory Data”. In: *IEEE Sensors Journal* 20.6 (2020). ISSN: 15581748. DOI: 10.1109/JSEN.2019.2956901.
- [152] Chao Ma et al. “IMU-based Locomotion Mode Recognition by Deep Convolutional Neural Networks with Wide First-layer Kernels”. In: *Proceeding - 2021 China Automation Congress, CAC 2021* (2021). DOI: 10.1109/CAC53003.2021.9727430.
- [153] Nattaya Mairittha, Tittaya Mairittha, and Sozo Inoue. “On-device deep personalization for robust activity data collection†”. In: *Sensors (Switzerland)* 21.1 (2021). ISSN: 14248220. DOI: 10.3390/s21010041.
- [154] Katrin Tomanek et al. *On-Device Personalization of Automatic Speech Recognition Models for Disordered Speech*. Ed. by Xiaoli Li et al. Vol. 1370. Communications in Computer and Information Science. Singapore: Springer Singapore, 2021. ISBN: 978-981-16-0574-1. DOI: 10.1007/978-981-16-0575-8.
- [155] Johannes Schneider and Michalis Vlachos. “Personalization of Deep Learning”. In: *Data Science – Analytics and Applications*. Wiesbaden: Springer Fachmedien Wiesbaden, 2021. DOI: 10.1007/978-3-658-32182-6_14.
- [156] Joel Shor et al. “Towards learning a universal non-semantic representation of speech”. In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH 2020-Octob* (2020). ISSN: 19909772. DOI: 10.21437/Interspeech.2020-1242.
- [157] Ramin Fallahzadeh and Hassan Ghasemzadeh. “Personalization without user interruption: Boosting activity recognition in new subjects using unlabeled data”. In: *Proceedings - 2017 ACM/IEEE 8th International Conference on Cyber-Physical Systems, ICCPS 2017 (part of CPS Week)* (2017). DOI: 10.1145/3055004.3055015.
- [158] Fatemehsadat Miresghallah et al. “UserIdentifier: Implicit User Representations for Simple and Effective Personalized Sentiment Analysis”. In: (2021).
- [159] Seunghyun Yoon et al. “Efficient transfer learning schemes for personalized language modeling using recurrent neural network”. In: *AAAI Workshop - Technical Report WS-17-01* - (2017).

- [160] Huy Thong Nguyen et al. “Personalization Models for Human Activity Recognition with Distribution Matching-Based Metrics”. In: *Communications in Computer and Information Science* 1370 (2021). ISSN: 18650937. DOI: 10.1007/978-981-16-0575-8_4.
- [161] Daniela Micucci, Marco Mobilio, and Paolo Napoletano. “UniMiB SHAR: A dataset for human activity recognition using acceleration data from smartphones”. In: *Applied Sciences (Switzerland)* 7.10 (2017). ISSN: 20763417. DOI: 10.3390/app7101101.
- [162] George Vavoulas et al. “The MobiAct dataset: Recognition of activities of daily living using smartphones”. In: *ICT4AWE 2016 - 2nd International Conference on Information and Communication Technologies for Ageing Well and e-Health, Proceedings Ict4awe* (2016). DOI: 10.5220/0005792401430151.
- [163] Kleomenis Katevas, Hamed Haddadi, and Laurissa Tokarchuk. “Poster: SensingKit - A multi-platform mobile sensing framework for large-scale experiments”. In: *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM* (2014). DOI: 10.1145/2639108.2642910.
- [164] Anna Ferrari et al. “On the Personalization of Classification Models for Human Activity Recognition”. In: *IEEE Access* 8 (2020). ISSN: 21693536. DOI: 10.1109/ACCESS.2020.2973425.
- [165] Daniel Roggen et al. “OPPORTUNITY: Towards opportunistic activity and context recognition systems”. In: *2009 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks and Workshops, WOWMOM 2009* February (2009). DOI: 10.1109/WOWMOM.2009.5282442.
- [166] Luis Sigcha et al. “Deep learning approaches for detecting freezing of gait in parkinson’s disease patients through on-body acceleration sensors”. In: *Sensors (Switzerland)* 20.7 (2020). ISSN: 14248220. DOI: 10.3390/s20071895.
- [167] Philipp M. Scholl, Matthias Wille, and Kristof Van Laerhoven. “A laboratory system for capturing and guiding experiments”. In: *UbiComp 2015 - Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (2015). DOI: 10.1145/2750858.2807547.
- [168] Attila Reiss and Didier Stricker. “Introducing a new benchmarked dataset for activity monitoring”. In: *Proceedings - International Symposium on Wearable Computers, ISWC* (2012). ISSN: 15504816. DOI: 10.1109/ISWC.2012.13.

- [169] Kerem Altun and Billur Barshan. “Human activity recognition using inertial/magnetic sensor units”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 6219 LNCS. 2010. ISBN: 3642147143. DOI: 10.1007/978-3-642-14715-9_5.
- [170] Jindong Wang et al. “Deep Transfer Learning for Cross-domain Activity Recognition”. In: (2018).
- [171] Federico Cruciani et al. “Personalizing activity recognition with a clustering based semi-population approach”. In: *IEEE Access* 8 (2020). ISSN: 21693536. DOI: 10.1109/ACCESS.2020.3038084.
- [172] Yonatan Vaizman, Katherine Ellis, and Gert Lanckriet. “Recognizing detailed human context in the wild from smartphones and smartwatches”. In: *IEEE Pervasive Computing* 16.4 (2017). ISSN: 15361268. DOI: 10.1109/MPRV.2017.3971131.
- [173] Luca Lonini et al. “Activity recognition in patients with lower limb impairments: Do we need training data from each patient?” In: *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS 2016-October* (2016). ISSN: 1557170X. DOI: 10.1109/EMBC.2016.7591425.
- [174] Melvyn Roerdink et al. “Evaluating asymmetry in prosthetic gait with step-length asymmetry alone is flawed”. In: *Gait and Posture* 35.3 (2012). ISSN: 09666362. DOI: 10.1016/j.gaitpost.2011.11.005.
- [175] Alexander Jamieson et al. “Human Activity Recognition of Individuals with Lower Limb Amputation in Free-Living Conditions: A Pilot Study”. In: *Sensors* 21.24 (Dec. 2021). ISSN: 1424-8220. DOI: 10.3390/s21248377.
- [176] James Gardiner et al. “Crowd-sourced amputee gait data: A feasibility study using YouTube videos of unilateral trans-femoral gait”. In: *PLoS ONE* 11.10 (2016). ISSN: 19326203. DOI: 10.1371/journal.pone.0165287.
- [177] Xiang Gao et al. “Knowledge-Guided Reinforcement Learning Control for Robotic Lower Limb Prosthesis”. In: *Proceedings - IEEE International Conference on Robotics and Automation* (2020). ISSN: 10504729. DOI: 10.1109/ICRA40945.2020.9196749.
- [178] Movesense. *Movesense Press Images*. URL: <https://www.movesense.com/press/> (Accessed on 01/13/2022).
- [179] Diane J. Cook, Narayanan C. Krishnan, and Parisa Rashidi. “Activity discovery and activity recognition: A new partnership”. In: *IEEE Transactions on Cybernetics* 43 (2013). ISSN: 21682267. DOI: 10.1109/TSMCB.2012.2216873.

- [180] Binbin Su, Yi Xing Liu, and Elena M. Gutierrez-Farewik. “Locomotion mode transition prediction based on gait-event identification using wearable sensors and multilayer perceptrons”. In: *Sensors* 21.22 (2021). ISSN: 14248220. DOI: 10.3390/s21227473.
- [181] Dalwinder Singh and Birmohan Singh. “Investigating the impact of data normalization on classification performance”. In: *Applied Soft Computing* 97 (2020). ISSN: 15684946. DOI: 10.1016/j.asoc.2019.105524.
- [182] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: <https://www.tensorflow.org/>.
- [183] François Chollet et al. *Keras*. <https://keras.io>. 2015.
- [184] Diederik P. Kingma and Jimmy Lei Ba. “Adam: A method for stochastic optimization”. In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* (2015).
- [185] *TensorFlow Lite — ML for Mobile and Edge Devices*. URL: <https://www.tensorflow.org/lite> (Accessed on 08/14/2022).
- [186] Feifan Cheng and Jinsong Zhao. “A novel process monitoring approach based on Feature Points Distance Dynamic Autoencoder”. In: *Computer Aided Chemical Engineering*. Vol. 46. 2019. DOI: 10.1016/B978-0-12-818634-3.50127-2.
- [187] Schalk Wilhelm Pienaar and Reza Malekian. “Human Activity Recognition using LSTM-RNN Deep Neural Network Architecture”. In: *2019 IEEE 2nd Wireless Africa Conference, WAC 2019 - Proceedings*. IEEE, Aug. 2019. ISBN: 9781728136189. DOI: 10.1109/AFRICA.2019.8843403.
- [188] Ottobock. *Empower — Ottobock UK*. URL: <https://www.ottobock.co.uk/prosthetics/lower-limb-prosthetics/prosthetic-product-systems/empower-ankle/> (Accessed on 12/15/2020).
- [189] Blatchford. *ElanIC - Waterproof Microprocessor Prosthetic Foot - Blatchford*. URL: <https://www.blatchford.co.uk/products/elanic/> (Accessed on 12/15/2020).
- [190] Ossur. *PROPRIO FOOT®*. URL: <https://www.ossur.com/en-gb/prosthetics/feet/proprio-foot> (Accessed on 12/15/2020).
- [191] Jana R. Montgomery and Alena M. Grabowski. “Use of a powered ankle-foot prosthesis reduces the metabolic cost of uphill walking and improves leg work symmetry in people with transtibial amputations”. In: *Journal of the Royal Society Interface* 15.145 (2018). ISSN: 17425662. DOI: 10.1098/rsif.2018.0442.

- [192] S Samprita et al. “LSTM-Based Analysis of a Hip-Hop Movement”. In: *2020 6th International Conference on Control, Automation and Robotics (ICCAR)*. IEEE, Apr. 2020. ISBN: 978-1-7281-6139-6. DOI: 10.1109/ICCAR49639.2020.9108052.
- [193] Angelo M. Sabatini et al. “Assessment of walking features from foot inertial sensing”. In: *IEEE Transactions on Biomedical Engineering* 52.3 (2005). ISSN: 00189294. DOI: 10.1109/TBME.2004.840727.
- [194] Michael Windrich et al. “Active lower limb prosthetics: A systematic review of design issues and solutions”. In: *BioMedical Engineering Online* 15.3 (2016). ISSN: 1475925X. DOI: 10.1186/s12938-016-0284-9.
- [195] Fan Zhang, Ming Liu, and He Huang. “Investigation of timing to switch control mode in powered knee prostheses during task transitions”. In: *PLoS ONE* 10.7 (2015). ISSN: 19326203. DOI: 10.1371/journal.pone.0133965.
- [196] Elisa Pedroli et al. *An Immersive Cognitive Rehabilitation Program : A Case Study : Proceedings of the 4th International Conference on NeuroRehabilitation (ICNR2018), October An Immersive Cognitive Rehabilitation Program : A Case Study*. Vol. 1. January. Springer International Publishing, 2019. ISBN: 9783030018450. DOI: 10.1007/978-3-030-01845-0.
- [197] Richa Sinha, Wim J.A. Van Den Heuvel, and Perianayagam Arokiasamy. “Factors affecting quality of life in lower limb amputees”. In: *Prosthetics and Orthotics International* 35.1 (2011). ISSN: 03093646. DOI: 10.1177/0309364610397087.
- [198] Hiram Ponce, Luis Miralles-Pechuán, and María De Lourdes Martínez-Villaseñor. “A flexible approach for human activity recognition using artificial hydrocarbon networks”. In: *Sensors (Switzerland)* 16.11 (2016). ISSN: 14248220. DOI: 10.3390/s16111715.
- [199] H. F. Maqbool et al. “A Real-Time Gait Event Detection for Lower Limb Prosthesis Control and Evaluation”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 25.9 (2017). ISSN: 15344320. DOI: 10.1109/TNSRE.2016.2636367.
- [200] Dongfang Xu et al. “Real-Time On-Board Recognition of Continuous Locomotion Modes for Amputees with Robotic Transtibial Prostheses”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 26.10 (2018). ISSN: 15344320. DOI: 10.1109/TNSRE.2018.2870152.
- [201] Rene Fluit et al. “A Comparison of Control Strategies in Commercial and Research Knee Prostheses”. In: *IEEE Transactions on Biomedical Engineering* 67.1 (Jan. 2020). ISSN: 15582531. DOI: 10.1109/TBME.2019.2912466.

- [202] Alex Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*. Vol. 385. Studies in Computational Intelligence 5. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. ISBN: 2000201075. DOI: 10.1007/978-3-642-24797-2.
- [203] Christopher Olah. *Understanding LSTM Networks [Blog]*. 2015. URL: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/> (Accessed on 06/24/2020).
- [204] Davide Anguita et al. “A public domain dataset for human activity recognition using smartphones”. In: *ESANN 2013 proceedings, 21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*. April. 2013. ISBN: 9782874190810.
- [205] Sebastijan Sprager and Matjaz B. Juric. “Inertial sensor-based gait recognition: A review”. In: *Sensors (Switzerland)* (2015). ISSN: 14248220. DOI: 10.3390/s150922089.
- [206] Saedeh Abbaspour et al. “A comparative analysis of hybrid deep learning models for human activity recognition”. In: *Sensors (Switzerland)* 20.19 (2020). ISSN: 14248220. DOI: 10.3390/s20195707.
- [207] Isibor Kennedy Ihianle et al. “A Deep Learning Approach for Human Activities Recognition From Multimodal Sensing Devices”. In: *IEEE Access* 8 (2020). ISSN: 2169-3536. DOI: 10.1109/access.2020.3027979.
- [208] Sakorn Mekruksavanich and Anuchit Jitpattanakul. “Smartwatch-based Human Activity Recognition Using Hybrid LSTM Network”. In: *Proceedings of IEEE Sensors 2020-October* (2020). ISSN: 21689229. DOI: 10.1109/SENSORS47125.2020.9278630.
- [209] Can Wang et al. “A flexible lower extremity exoskeleton robot with deep locomotion mode identification”. In: *Complexity* 2018 (2018). ISSN: 10990526. DOI: 10.1155/2018/5712108.
- [210] Akbar Dehghani, Tristan Glatard, and Emad Shihab. “Subject Cross Validation in Human Activity Recognition”. In: *CoRR* abs/1904.0 (Apr. 2019).
- [211] Daniel Roggen et al. “Collecting complex activity datasets in highly rich networked sensor environments”. In: *2010 Seventh International Conference on Networked Sensing Systems (INSS)*. IEEE, June 2010. ISBN: 978-1-4244-7911-5. DOI: 10.1109/INSS.2010.5573462.
- [212] Timo Sztyler and Heiner Stuckenschmidt. “Online personalization of cross-subjects based activity recognition models on wearable devices”. In: *2017 IEEE International Conference on Pervasive Computing and Communications, PerCom 2017* (2017). DOI: 10.1109/PERCOM.2017.7917864.

- [213] Xavier Glorot and Yoshua Bengio. “Understanding the difficulty of training deep feedforward neural networks”. In: *Journal of Machine Learning Research* 9 (2010). ISSN: 15324435.
- [214] Md Zia Uddin et al. “Estimation of mechanical power output employing deep learning on inertial measurement data in roller ski skating”. In: *Sensors* 21.19 (2021). ISSN: 14248220. DOI: 10.3390/s21196500.
- [215] Gang Du et al. “Locomotion Mode Recognition with Inertial Signals for Hip Joint Exoskeleton”. In: *Applied Bionics and Biomechanics* 2021 (2021). ISSN: 17542103. DOI: 10.1155/2021/6673018.
- [216] Manuel Andrés Vélez-guerrero, Mauro Callejas-cuervo, and Stefano Mazzoleni. “Artificial intelligence-based wearable robotic exoskeletons for upper limb rehabilitation: A review”. In: *Sensors* 21.6 (2021). ISSN: 14248220. DOI: 10.3390/s21062146.
- [217] Wan Shi Low et al. *A Review of Machine Learning Network in Human Motion Biomechanics*. Vol. 20. 1. Springer Netherlands, 2022. ISBN: 0123456789. DOI: 10.1007/s10723-021-09595-7.
- [218] Tarek M. Bittibssi et al. “Implementation of surface electromyography controlled prosthetics limb based on recurrent neural network”. In: *Concurrency and Computation: Practice and Experience* June 2021 (2022). ISSN: 1532-0626. DOI: 10.1002/cpe.6848.
- [219] Movesense. *Movesense — Technical Specifications*. URL: <https://www.movesense.com/wp-content/uploads/2020/12/Movesense-Spec-Sheet-12-2020.pdf> (Accessed on 07/23/2022).