INVESTIGATION OF LOW-COST WEARABLE INTERNET OF THINGS ENABLED TECHNOLOGY FOR PHYSICAL ACTIVITY RECOGNITION IN THE ELDERLY

A thesis submitted in partial fulfilment of the requirements of Liverpool John Moores University for the degree of Doctor of Philosophy

Lee Stephen Newcombe

May 2020

Table of Contents

List of Figures

List of Equations

List of Tables

Acknowledgements

I would like to express my thanks to my supervisors, Dr. Po Yang, Dr. Chris Carter and Dr. Martin Hanneghan for their insight, suggestions and support over the course of this PhD project. Their aid throughout this process has been invaluable and greatly appreciated. I would also like to thank Dr. Po Yang for providing his advice and guidance over the course of the project and for pushing me to achieve a variety of research approaches and for building my confidence in my presenting ability. I would like to thank Dr. Chris Carter for his advice in achieving a working solution for my thesis.

I would also like to thank my friends for their support over the course of the project. Their help in addressing problems with data collection and participating in trial runs of the process is greatly appreciated. I would also like to thank the staff in the Faculty Research Office for their support over the last few years.

I would also like to thank Join Dementia Research for their assistance and support for my feasibility study. Finally, I would like to thank my parents for their constant support. They have always encouraged me to push myself and it is with their support that I was able to complete this project.

Abstract

Technological advances in mobile sensing technologies has produced new opportunities for the monitoring of the elderly in uncontrolled environments by researchers. Sensors have become smaller, cheaper and can be worn on the body, potentially creating a network of sensors. Smart phones are also more common in the average household and can also provide some behavioural analysis due to the built-in sensors. As a result of this, researchers are able to monitor behaviours in a more naturalistic setting, which can lead to more contextually meaningful data. For those suffering with a mental illness, non-invasive and continuous monitoring can be achieved. Applying sensors to real world environments can aid in improving the quality of life of an elderly person with a mental illness and monitor their condition through behavioural analysis. In order to achieve this, selected classifiers must be able to accurately detect when an activity has taken place.

In this thesis we aim to provide a framework for the investigation of activity recognition in the elderly using low-cost wearable sensors, which has resulted in the following contributions:

1. Classification of eighteen activities which were broken down into three disparate categories typical in a home setting: dynamic, sedentary and transitional. These were detected using two Shimmer3 IMU devices that we have located on the participants' wrist and waist to create a low-cost, contextually deployable solution for elderly care monitoring.
2. Through the categorisation of performed Extracted time-domain and frequency-domain features from the Shimmer devices accelerometer and gyroscope were used as inputs, we achieved a high accuracy classification from a Convolutional Neural Network (CNN) model applied to the data set gained from participants recruited to the study through Join Dementia Research.

The model was evaluated by variable adjustments to the model, tracking changes in its performance. Performance statistics were generated by the model for comparison and evaluation. Our results indicate that a low epoch of 200 using the ReLu activation function can display a high accuracy of 86% on the wrist data set and 85% on the waist data set, using only two low-cost wearable devices.

List of Abbreviations

IoT	Internet of Things
PA	Physical Activity
PAR	Physical Activity Recognition
PAM	Physical Activity Monitoring
AR	Activity Recognition
SH	Smart Home
BA	Behaviour Analysis
ADL	Activities of Daily Living
AAL	Ambient Assisted Living
ANN	Artificial Neural Networks
CNN	Convolutional Neural Network
DT	Decision Tree
SVM	Support Vector Machine
MCI	Mild Cognitive Impairment
QoL	Quality of Life

Chapter I: Introduction

1.1. Background

It is currently estimated that 1 in 5 people over the age of 65 will develop dementia, with an estimated 850,000 currently living with some form of the disease [1]. In order to measure and conceptualize Dementia related behavioural symptoms, it is essential that the disease is understood. Dementia has two major symptom groups: cognitive dysfunction symptoms as well as symptoms of behavioural and psychological signs [2]. The behavioural symptoms of the disease include combinations of changes in physical movement and speech. One such example of a change to a dementia person's behaviour is agitated behaviour [3]. Speech ability in a person suffering from dementia can cause communication challenges, however maintaining effective communication increases the quality of life for the person as persons suffering from dementia can frequently struggle to find the appropriate words to describe objects. Various behavioural symptoms of dementia such as depression and anxiety can be caused by the difficulties that persons suffering with dementia have with communication [4].

Alternative occurrences of behavioural dementia symptoms include sleep disturbances, withdrawal, and apathy. Behavioural dementia symptoms can vary between individuals in repeated occurrences as well as the symptoms that become present in a person. Clusters of behavioural symptoms can also occur in a person which can make characterizing the symptoms significantly more complicated [3]. In some circumstances, clinical applications monitor behaviour through direct observation of the person. However these applications only consider whether the behaviour is present or absent, only few applications consider the intensity of the behavioural symptoms that are present [3].

Behaviour-change approaches typically involve the direct observation of real-life settings and behaviour change principles to enhance the quality of life of people who suffer with dementia. Presently, there are several theoretical and computational frameworks for modelling dementia-related behavioural excesses (wandering, disruptive vocalisations), behavioural deficits (incontinence, self-feeding) and mood changes (depression) for critical care and hospital care environments [5]. However, in order to use these frameworks, large-scale ambient sensing systems have to be deployed which are expensive to implement and only viable in a hospital care environment.

Pervasive healthcare applications have become fairly common over the past decade [6], with many applications taking advantage of recent technological developments in the mobile and wearable device market. Developers of these devices are continually improving upon the technology used to add more processing power, storage space and even adding new sensors into these hand-held and wearable devices. Pervasive healthcare applications developed for these platforms can be used to deliver assistance to those with disabilities such as communication issues, or to monitor and evaluate the behaviour of a person. In comparison to other diseases, Dementia is one such disease where the carers can also benefit from healthcare applications on smart devices. These applications can function as an assistive application, providing prompts to alert the user to take their medication, or provide instructions on how to perform various tasks. These applications can also take on a Human-Computer Interaction (HCI) role and make use of the sensors in the device to monitor cognitive decline or behaviour. Monitoring how a person performs a task, such as taking a drink of water, can be crucial in evaluating how fast the disease progresses in different people.

The advances in Internet of Things (IoT) related technologies over the last few years have provided new opportunities to build Quality of Life (QoL) profiles of an individual with increasing validity and reliability [7]. This has become possible by monitoring the lifelogging data captured by a variety of IoT technologies (sensors, mobile apps, web-objects, etc.) with continuous connectivity and interaction in a pervasive network. Presently, those with long-term conditions and chronic diseases require intense interactions with clinicians in a hospital environment. This can be time consuming, and the resulting assessment can be subjective. It can also be costly to the hospital and therefore not sustainable [8]. Utilising IoT technology for home-based Dementia care will provide more accurate monitoring and deep analysis of dementia related behaviour in a home-based environment such as: gradual loss of memory, difficulty in performing familiar or complex tasks, changes in mood and disorientation. IoT technologies also allow us to consider sensitivity, social and emotional factors such as working with persons that are at various stages of the disease.

Currently there is little evidence-based literature for guiding the implementation of strategies in order to ensure an early diagnosis or to design an optimal service provision for people that suffer with the disease. The aim of this thesis is to collate various papers of research and systems in order to provide insight into what technologies are currently available in the field. We will address our survey findings and provide a discussion on the advantages and disadvantages of existing technologies. This will allow us to provide our thoughts on future directions research could take in this field.

1.2. Research Aim and Objectives

The following section highlights the main aims and objectives of this research project. Research novelties are also outlined in this section.

a) Aims

This research aims at investigating Internet of Things enabled technologies for delivering a new wearable healthcare assistive solution for effectively and efficiently monitoring, analysing and understanding complex home-based dementia patients' behaviours. The project involves the design of a generic Internet of Things enabled mobile health framework to leverage the potential of mobile devices like smartphones or tablets, wearable sensors, for monitoring and analysing dementia related behaviour. This framework will also provide algorithms for simultaneous and long-term quantification of behaviour change in dementia persons with IoT enabled wearable devices. This new wearable healthcare assistive technique will also potentially identify and mitigate any issues related to poor information transference as well as interpretation (*human-to-human and system-to-human*), process management, cognitive capacity, and patient related needs.

b) Objectives

The main objectives for this project are:

1. To explore IoT methods to study and classify activity for people who suffer with dementia.
2. To design a low-cost wearable device framework suitable for the needs of the elderly and sufferers of dementia to perform physical activity recognition (PAR) in uncontrolled environments.
3. To evaluate algorithms that will be capable of classifying activities that are performed in a home-based environment.
4. To carry out a thorough evaluation of the delivered system to validate its technical capacity and to examine the potential impact it will have on future healthcare by working closely with the end users including carers and patients.

1.3. Thesis Outline

In order to achieve the research aims and objectives of the project, we have conducted an extensive survey of literature and provided four chapters which contain the contributions of this project. This includes a framework for the PAR of elderly people, PA recognition in the participants with no known medical conditions and PA recognition in elderly participants that have memory loss or an official diagnosis of mild cognitive impairment. The outline of the thesis is as follows:

Chapter 2 provides an extensive literature review and introduces the approaches taken in human physical activity recognition using IoT technologies. The main components of IoT, sensor layer, processing layer and application layer are studied and have the main techniques in wearable and ambient sensing analysed.

Chapter 3 discusses our framework for physical activity recognition in the elderly using low-cost devices. Activity types that are performed by participants are discussed and categorised. Wearable sensors are identified and discussed, with potential positioning being an important factor in order to ensure data collection from specific PA.

Chapter 4 presents a PAR study using healthy participants to perform 15 activities in a controlled environment. Data collection, pre-processing techniques and feature extraction techniques used are introduced with a discussion on the raw signal information we collected.

Chapter 5 introduces three classifiers used on the data collected. ANN, SVM and DT classifiers were detailed, and were run using the data collected as the input. The results from the classifiers is detailed, along with an evaluation of performance. Furthermore, we present a CNN model using 1-dimensional convolutional layers that we will implement in our main study.

Chapter 6 introduces our main study of the project where we collected data from 3 participants for 18 activities. Participant recruitment is discussed along with the data collection processes. The CNN model developed is used to classify the activities from the collected data. The model is then evaluated based on its performance achieved from a series of tests using various activation functions and parameter changes.

Finally, we present a summary of the project in chapter 7. Limitations of the project are highlighted along with future endeavour. Conclusions of the project are also drawn in this chapter.

1.4. Research Contributions

This thesis provides the following contributions:

- An extensive and systematic review and analysis of PAR studies from an IoT perspective. Traditional and state-of-the-art PAR methods used in the healthcare domain, including sensors and recognition techniques are discussed and summarised.
- A novel framework for investigating physical activity recognition in the elderly. Activity types are identified based on their suitability in a home environment. Potential classifiers are highlighted for investigation.
- A PAR study collecting data from elderly participants that are suffering with memory loss or have received a diagnosis of Mild Cognitive Impairment. Participants performed activities in a controlled environment using low-cost wearable devices on the wrist and waist during data collection.
- A CNN model is developed and deployed for classifying the collected data. An evaluation into the performance of this model is provided. The results of the study indicate a low epoch using a ReLu activation function provides a high accuracy and a low loss.

1.5. List of Publications

During the course of this research, the following research has been published, including four conference papers and two journal publications, alongside various paper presentations of this research:

Conference Papers:

L. Newcombe, P. Yang, C. Carter, M. Hanneghan and J. Qi, "Experimental analysis of cost-effective mobile sensing technologies for activity analytics in elderly care", 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), **Pages: 1442 – 1448**, 28-30 June 2018, Exeter, UK.

DOI: 10.1109/HPCC/SmartCity/DSS.2018.00238

L. Newcombe, P. Yang, C. Carter and M. Hanneghan, "Internet of Things Enabled Technologies for Behaviour Analytics in Elderly Person Care: A Survey", 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and

Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), **Pages: 863 – 870**, 21-23 June 2017, Exeter, UK.

DOI: 10.1109/iThings-GreenCom-CPSCom-SmartData.2017.133

L. Newcombe, P. Yang, C. Carter, M. Hanneghan, A. Simpson and J. Qi, "A Comparison Study of Wearable Sensing based Classifier Algorithms for Aging Behaviour related Physical Activity Recognition", **41st Engineering in Medicine and Biology Conference 2019**, Accepted, 2019.

J. Qi, X. Peng, **L. Newcombe**, A. Simpson, Y. Yang and P. Yang, "Experimental Analysis of Artificial Neural Networks Performance for Home-based Physical Activity Recognition Using Belt and Wristband Devices", **41st Engineering in Medicine and Biology Conference 2019**, Accepted, 2019.

Journal Papers:

J. Qi, P. Yang, **L. Newcombe**, X. Peng, Y. Yang and Z. Zhao, "An Overview of Data Fusion Techniques for Internet of Things enabled Physical Activity Recognition and Measure", Journal: Information Fusion (**IF 6.6**), Accepted 2019, https://doi.org/10.1016/j.inffus.2019.09.002.

J. QI, X. Peng, Y. Yun, **L. Newcombe**, G, Yang, Y. Liu and P. Yang, "A Hybrid Hierarchical Model for Accessing Physical Activity Recognition towards IoT enabled Free-living Environments", Journal: IEEE Internet of Things Journal (**IF 9.5**), Under Review.

Conference Presentations:

Paper Presentation, 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 21-23 June 2017, Exeter, UK.

Paper Presentation, 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 28-30 June 2018, Exeter, UK.

Paper Presentation, 2019 IEEE 41st Engineering In Medicine and Biology Conference, 23-27 July 2019, Berlin, Germany.

1.6. Summary

Physical activity recognition and monitoring provides important benefits to the quality of life to the elderly population and can also have lasting effects on mental and physical health. The use of IoT technologies in this area has become widespread in many research studies and rehabilitation programs. However, a majority of these studies are performed in a hospital setting and as a result can only provide subjective results. Sensors used can also be extremely expensive, however advances in sensor technology have allowed cheaper wearable alternatives to become a focus for further research while maintaining good results. These studies are typically controlled and in a lab environment which can provide only a limited set of data. A focus on providing PAR from a home environment could provide more reliable data and could also be scaled to allow for more participants. Another limitation of current studies is the diversity and complexity of activities. This also cannot be fully expressed by participants in a controlled environment and can lead to a limited data set. By allowing studies to take place at home, researchers can expand the range of activities performed, while also capturing more reliable and natural activity data. Maintaining a high activity recognition accuracy while also keeping costs to a minimum and alleviating privacy concerns in home environments is a challenge that researchers must tackle.

In this thesis we provide an extensive review into previous PARM studies from an IoT perspective. Focusing on the state-of-the-art methods used in the healthcare domain. We provide a design for a framework for the investigation of AR in the elderly population. The framework is designed to use low-cost wearable devices and smart phones that could allow for a small lab-based experiment be scaled into a home-based study allow for the integration of a diverse range of activity types for data collection. We produced a CNN model using 1-dimensional Convolutional layers and dense layers to classify chosen activities that were performed in an activity recognition study on elderly people, that display signs of memory loss or have a diagnosis of Mild Cognitive Impairment.

The model was tested using the activity data collected from participants with various activation functions and epochs in order to provide an evaluation of the model's performance. We found that our model can produce relatively high accuracy on activity classifications and did so in a suitable time period despite not running on a GPU. This could make our model suitable to a clinical environment were the model might need daily executions by clinicians.

Chapter II: Literature Review

2.1. Introduction

In order to understand how the advancements made into IoT technologies have improved our ability to recognise PA, we must conduct a review into the various approaches used in PAR. We will begin this survey by observing the various layer-based approaches to PA using IoT technologies as shown in Figure 2-1. We will also observe how these approaches are used in recognising PA of elderly people and people with Dementia and if the technology available can be used for behaviour analysis. For this survey, papers were gathered from the following online libraries; IEEE Explore, ACM Digital Library, Science Direct and Google Scholar.

The following keywords and phrases were used to perform the survey; Behaviour analysis using smart devices in the elderly, wandering detection in the elderly, agitation and anxiety detection using wearable sensors in the elderly, monitoring depression with smartphones, fall detection in the elderly using smartphones and detection of agitation using smart devices. Our search generated a large number of results relating to PAR in elderly people and in healthcare as shown in Figure 2-2. The main goal of this literature review is to classify the IoT technologies observed in terms of a system architecture, to review current IoT technologies that are involved in PAR and to review PAM in healthcare.

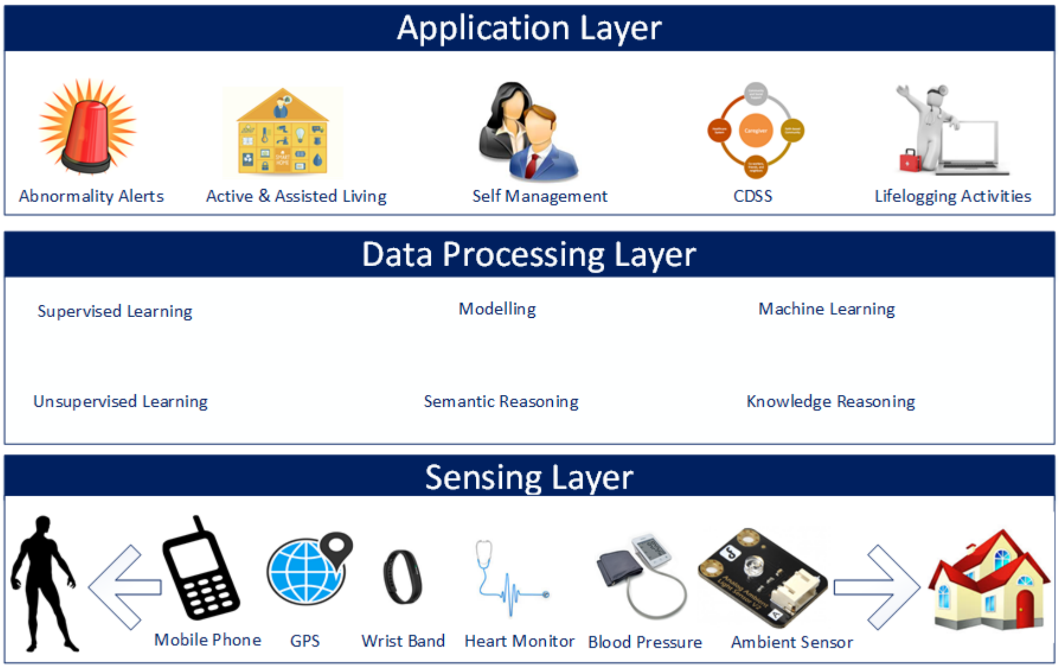


Figure 2-1: IoT layer-based approaches to PAR

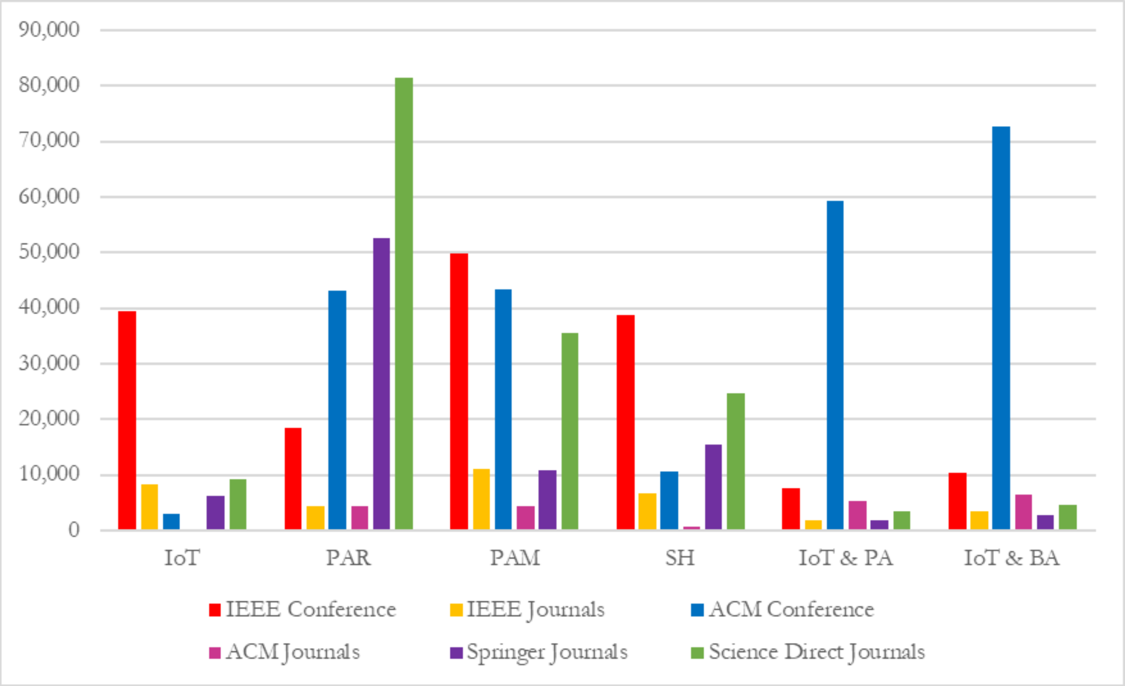


Figure 2-2: Number of Journals and Conference Articles relating to IoT technologies in PAR between 2010-2019

2.2. Sensing Layer

The sensing layer involves employing singular or multiple sensors, either on the body or in the environment in order to gather data relating to PA through lifelogging. The process of lifelogging refers to the recording of an individual's life using digital devices to monitor physical activity or to provide medical interventions[9]. Initially, lifelogging was performed using a camera to capture video or images. This, however, could be construed as an invasion of privacy and has become less popular as new IoT technologies have developed [10], [11]. Wearable technology has become a popular solution for lifelogging as it is not invasive and can contain multiple sensors in one device, providing a greater range of results. However, many of these devices can be expensive and as a result are typically used by researchers and healthcare professionals in a controlled environment. Devices that are cheaper to use, such as a smartphones, typically produce less accurate results. The data produced by the devices can be large, so storage requirements are another aspect of PAM that has to be considered.

The advancements in low cost IoT wearable devices has led to improvements in accuracy when monitoring PA in home environments. This also allowed researchers to build quality-of-life profiles of individuals with increasing reliability and validity [7]. As IoT technologies can be used in monitoring PA in home-based environments, we can monitor the PA of elderly people and people with Dementia. In terms of Dementia, we must understand the related symptoms and the behaviours associated with the disease. These are typically classified under 6 neuropsychiatric symptoms [12]: 1) Anomalous Motor Behaviour, 2) Depression, 3) Anxiety, 4) Weight Loss, 5) Irritability, 6) Agitation. Monitoring these symptoms is possible through wearable technology. Utilising IoT technologies can allow us to consider sensitivity, social and emotional factors of working with elderly people and people with Dementia [7].

2.2.1. Physical Activity

According to the World Health Organisation, 60% of the global population fails to reach the recommended 30 minutes of daily intense physical activity [13]. This can lead to health risks in both the developed and developing world. Reasons for this lack of activity can be found in the advances in technology that have reduced the amount of PA needed to perform activities in work and at home. Attempting to achieve the recommended amount of PA can help in managing weight and reducing blood pressure. This can also affect muscle strength and posture stability, two factors that can influence the frequency of falls in elderly people [14].

Table 2-1: Different Activities types that can be used in PAM

	Types of Activity	Activities
Simple Activities	Sedentary Activities	Sitting, Standing, Lying
	Transitional Activities	Sit-to-Stand, Stand-to-Sit, Sit-to-Lie, Lie-To Sit
	Aerobic Exercise	Walking, Jogging, Running, Swimming
Complex Activities	Activity of Daily Living	Drinking Water, Brushing Teeth, Eating, Getting Dressed
	Weight Lifting	Bench Press, Deadlifts, Squats

Activity recognition is a method that can be employed in order to monitor simple and complex PA that a person can perform. Table 2-1 displays examples of activity types that can be recorded. ADL is an example of the types of activities that are typically monitored in elderly person care. However PAM can be used in preventative healthcare in order to change a person's behaviour in order to reduce the risk of the individual contracting an illness [14].

An example of elderly activity monitoring using wearable devices is provided by [15]. The device used in their service is a Bluetooth smart beacon that has accelerometer, gyroscope and Bluetooth functionality. Acceleration and angular velocity is collected and transmitted via Bluetooth to a smartphone with a health monitoring app. The data is then sent to the cloud for further processing where the activity life log can then be accessed by a care giver. Six subjects between the ages of 22 and 30 were used to perform a series of activities such as drinking, washing hands and using the bathroom fifty times. The data produced was split into 60% training and 40% testing and was classified using an REP tree [15].

An activity recognition system using a series of tri-axial accelerometers to monitor the daily activities in a home environment was proposed by [16]. Accelerometers were placed on the wrist, thigh and arm. The sampling frequency of the accelerometer was set to 20Hz and the output range was ±6g. A group of participants performed a series of running, walking, sitting and climbing activities. The wireless unit attached to the sensor communicated with an IoT cloud server to transmit and store the data collected for analytics. A Naïve Bayes machine learning package was installed on the cloud server for processing the data in order to classify the activities. The result of processing produced an accuracy of 72% for the system.

The IDAct system, proposed by [17] utilises passive long-range RFID sensors on physical objects. The system can be used for food and health monitoring by detecting what food a user is preparing or when they are taking medication. IDAct can also be combined with assistive technologies to improve the QoL of elderly people through PAR. A single RFID reader was used to collect data from 110 tagged objects that were located in a home environment that consisted of four living spaces; the kitchen, the bathroom, a dining room and a living room. 10 participants were asked to perform 24 daily activities including cooking and watching TV. Video recordings were also captured to evaluate the system. An HMM classifier is used to train and test the data collected from sensors to achieve an accuracy range of 46% to 100% across the 24 activities.

2.2.2. Anomalous Motor Behaviour

Anomalous Motor Behaviour (AMB) refers to the behaviours exhibited by physical movement. This can include location-based behaviours such as wandering which can be monitored through GPS and verbal behaviours can be viewed as motor behaviours. In order to monitor AMBs a range of sensor technologies can be implemented ranging from non-medical grade sensors to blind video monitoring [12].

Lifelogging the physical activity (PA) of an elderly person is vital for monitoring the health of a person. Initially attempts to capture data were performed by an external camera; however, this was considered an invasion of privacy [9]. Over recent years, various wearable trackers have been developed to monitor physical activity. Sensors such as, FitBit, Nike+ Fuelband, are gaining more attention publicly as they can record information such as heart rate and calories burnt [9]. Due to the heterogeneous nature of data sets, lifelogging PA data is generally more challenging to handle. Traditional methods require machine learning algorithms and sensors to analyse the PA, activity patterns and level of intensity. However, these methods analyse human behaviours from raw sensor data [18], [19]. With the purpose of maintaining a high accuracy when logging PA data, sensors may have to be worn on the body. These are not often cost-effective solutions in real environments [20].

Aguiar [21] presents an accelerometer-based approach for fall detection using smart phones. The paper mentions the necessity for monitoring fall detection as 40% of all mortality in older persons is caused by falls. It has also been discovered that a previous faller has a two-thirds probability of falling again within the next year. As a result, reliable fall detection and emergency assistance notification are required in order to provide suitable care and to increase the quality life of the elderly. The application itself is designed for android smartphones and aims to distinguish falls and typical activities of daily life (ADLs).

Aguiar [21] has designed the application to run as a background task on the smartphone, in order to monitor all of the user's movements to detect a fall. The system will sound an alarm if a fall is detected as well as providing an SMS message to a set list of contacts. The proposed solution for the application considers an algorithm based on a state machine. The algorithm recognizes the stages of a fall and will trigger an alarm if the stages are recognized in the correct order. Transitioning between stages is performed by monitoring changes in the computing signals, using offline classification algorithms to obtain thresholds

Table 2-2: Internet of Things Enabled Technology for Monitoring AMB in Elderly People

Type of Symptoms	Sensors	Algorithms	Applications	Advantages	Disadvantages
Wandering	Wrist Worn Activity Monitor [18], [19], [22]	Algase Wandering Scale [22]	Wandering Behaviour Framework [22]	Wrist worn devices can provide constant activity monitoring.	Wrist worn sensors could be considered an irritant.
	Step Counter [22]	Cycles Algorithm [23]	Endomondo [18], [19]	Step sensors could determine wandering based on a constant increase of steps.	Sensor is required to be worn and could be forgotten or be considered irritating.
	Heart Rate Monitor [22]	Activity Pattern Recognition [24]	Moves [9], [18]	Heart monitors can monitor increases in heart rate to determine activity.	Heart rate can increase and decrease due to a number of factors, leading to inconsistent results
	GPS [23], [25], [26]	Eulerian Cycles [25]	iWander Application [26]	GPS allows the movements of a person to be easily monitored	Requires the devices to be on the person at all times. Could provide inconsistent data if forgotten.
	Tri-Axial Accelerometer [24], [27]		SIMPATIC [25]	Increased speed in data collection. Produces results along all three axes.	Device has to be varied on the person's body. Can be forgotten or could lose connection if device is wired.
Falling		Tri-Axial Accelerometer Algorithm [27]			
	Floor Sensor [28]	Event Detection and Segmentation [28]	Fall Detection Application [29]	Can accurately detect a person falling on the floor.	Could detect other objects falling onto the floor, providing inaccurate results.
	Accelerometer [21], [29]–[31]	Discrete Wavelet Transform [29]	Android Smartphone Application [21]	Can detect sudden movements involved in a fall.	Device could be dropped, producing a fall result.
	Pulse Sensors [30], [32]	State Machine Based Algorithm [16]	Pulse Wave Smartphone Application [32]	Pulse sensors can notice changes in heart rate, which could signal a fall.	Device can be forgotten by person, leading to a gap in data collection. Other factors can also cause changes in heart rate.

The method proposed by Kim [24] uses tri-axis acceleration sensors to digitalize the behaviour of the elder by monitoring daily activities. Activities are categorised as Low and High level. Low level refers to physical movements such as walking, running and lying. This data can be detected with sensors as they can recognize when the subject moves and how long the subject has moved. High-level activities are predicted by time and place, for example sleeping. This information is unreliable, as it requires some guesswork on the part of the researcher as it is difficult to recognise the activity through the raw sensor data.

Yavuz [29] proposes a fall detection system for android smartphones that incorporates different algorithms and accelerometers on the phone in order to create a robust fall detection system. Yavuz suggests that using a 1-D Fourier frequency is not sufficient for detecting falls using accelerometer signals. The alternative proposed is the wavelet transformation method as this allows for temporal localization of frequency components.

Discrete wavelet transform (DWT) is able to yield a multi-scale representation of discrete signals, which are formed by applying analysis filters to the original signal. For their application, Yavuz [29] uses DWT as a feature extraction method. The DWT method is applied to a discrete acceleration signal provided by the accelerometer integrated into the phone. If the values of the coefficients that are extracted from the accelerometer are above the threshold, then a fall is detected. The application is designed for users who may be susceptible to sudden falls like the elderly, those with epilepsy or mild cognitive impairment such as dementia sufferers. It allows alerts to be sent to carers as well as to contacts over social media should a sudden fall occur.

Fall detection solutions can have their disadvantages. One example being the "Social Alarm" [28] as this requires the user to press a button. After a fall, a person could be unconscious or panicked and as a result unable to press the button to alert a carer. The most popular solution, the wearable fall detectors that are based on accelerometers and tilt sensors, require the person to wear the device constantly, such as in the shower where the risk of a fall is greatest [28].

Video based analysis techniques rely on image processing of a person's movements in real time. The video system must be able to detect objects, extract features such as speed and be able to determine if a human fall has occurred. However, this presents challenges, as it can be difficult to monitor an entire house and this technique has a number of privacy concerns [28]. However, in situations where it can be deemed appropriate, video can provide more information than simple motion related data. The example given in [12] is the problem of ensuring elderly people have sufficient nutrition as they can sometimes forget to eat. In this situation, video capture could be used to monitor body mass changes.

2.2.3. Anxiety

Miranda et al. [33] proposes an application that detects anxiety through wearable sensors. Three devices are employed to monitor heart rate and spontaneous blink rate. The custom application was developed to transfer data produced by Google Glass and a Bluetooth zephyr HxM band to a server. The set-up involved a dedicated router to reduce transfer delays and other common issues found in scholar networks in order to prevent data loss during data transfer between application and server. Miranda et al. coded their algorithm in Java based on the approach by Ishimaru [34] with some minor modifications. Sensor readings were tracked to detect when data hits a peak value to identify the blink event. When a peak was detected, it would be recognised as a blink.

The Google Glass application proposed by [33] allowed data to be gathered by the IR sensor in order to detect agitation through blink rate. The IR sensor measures the distance between the device and the user's eye and can determine a blink event as a result. However, the disadvantage of this could be due to the requirement to wear the Google Glass device constantly in order to establish when the person is agitated, and this requirement itself could agitate the user. This method could also be seen as unreliable as the user could experience sudden relaxation which could affect results, as pointed out in [33].

Bankole et al.[35] uses body sensor networks for continuous longitudinal agitation assessment. A multi-trait-multi-method approach is used to test the validity of the body sensor network. Three clinical instruments (Aggressive behaviour scale, Cohen-Mansfield Agitation Inventory, Mini Mental State Examination) with strong validity and reliability were used as benchmarks to test convergent and discriminant validity of the BSN.

The TEMPO3 network proposed by Bankole [35] consists of up to eight wireless inertial body sensors that gather data from persons while they perform normal daily activities. The motion capture capabilities provided by TEMPO allow for six degrees of freedom, sensing at arbitrary frequencies with 12-bit resolution. This enables higher resolution analyses of persons' movements and potentially higher precision assessments of agitation, as it is essential to differentiate normal movement from agitated movement. Battery life on the devices is limited to ~5 hours when all six sensors are streaming the raw data via a Bluetooth transceiver. When the gyroscopes are turned off, the battery life increases to ~9 hours. This is done as only the accelerometers are used in the study by Bankole [35]. Local storage of the data instead of transmitting it wirelessly increases the battery life further, which would be permitted in a clinical deployment of the system.

Lu et al. [36] studied stress associated with the cognitive load experienced by a participant during a job interview as an interviewee and conducting a marketing task as an employee. Neutral tasks were also considered, where participants are not under stress. These three tasks are designed with the help of behavioural psychologists. As with SUSAS (Speech under Simulated and Actual Stress) and other previous studies [37]–[39], it is assumed that the subject's voice is stressed once the stressor is present and reading without the stressor is neutral. Audio was continuously collected using a Google Nexus One Android smartphone and a microcone1 microphone array. In addition to audio data, video cameras record the interviewer and interviewee. In addition to capturing stressed audio, we also collect audio data from neutral scenarios where participants are not stressed. In neutral scenarios, participants had to read both indoors and outdoors. The reading materials are simple stories that are often used to study different accents in languages.

Table 2-3: Internet of Things Enabled Technology for Monitoring Anxiety and Agitation in Elderly People

Sensors	Algorithms	Applications	Advantages	Disadvantages
Heart Monitor [33]	Blink Event Algorithm [33]	Google Glass Application [33]	Heart monitors can monitor increases in heart rate to determine activity.	Heart rate can increase and decrease due to a number of factors, leading to inconsistent results
Body Sensor Network [35]	Peak Detection [40]	StressSense [36]	A sensor network can perform constant monitoring of activities.	Relies on the person to remember to wear the sensors. Persons could be irritated by wearing devices constantly.
Video Monitoring [12]	Joint Time-Frequency Algorithms [35]	TEMPO3 [35]	Video monitoring is accurate in detecting signs of agitation.	Privacy concerns can outweigh the benefits from using video monitoring.

2.2.4. Depression

Gruenerbl et al. [41] gave each participant an Android smart phone that ran a logging application developed by their group. The application contained two major components; a data logger (using the standard Android API) with the person having the option of turning off the logging at any time and a self-assessment questionnaire (set to pop up at the end of the day). After finishing the questionnaire, the person would be asked whether they were comfortable with logging the day's data. The data would be stored on an SD card and was copied during the periodic examination and stored in a form that would not reveal the person's identity to the researchers working on the data later on. Clearly, in a "productive" system the data would need to be transmitted wirelessly at the end of each day. However, for the purpose for the research by Gruenerbl et al. [41] the SD card option was more reliable and allowed us to simplify data security issues.

Canzian [42] uses the Android application MoodTraces that automatically samples phone sensors to retrieve the current location, which is represented by a time reference, a longitude value, and a latitude value. Additional information about the phone usages and user activities extracted using the Android API are also collected, but they are not analysed directly in this work, given its specific focus on mobility pattern analysis. Activity information is also used to optimize the sampling process. In addition to sensor-based data, MoodTraces collects the answers that the users provide to daily questionnaires.

Burns [43] used a single-arm field trial of Mobilyze, an 8 week multimodal intervention for depression that included 1) mobile phone sensing and ecological momentary intervention 2) an interactive website for behavioural skills training and 3) email and telephone support from a coach assigned to each participant. The context aware system used an architecture consisting of 3 phases. Phase 1 uses sensors that are housed on the mobile phone to perform observations of the participants and their environment. Phase 2 used an algorithm to inductively learn the relationship between the sensor data and the participants recorded social context, activity, location and internal states. Phase 3 consisted of action components that provided mechanisms for relaying predictions to other external outreach applications.

There are several sleep monitoring systems on the market to date that can aid in the detection of depression [44]. Polysomnography devices are the most accurate however, the drawbacks to these devices include the expensive price tag as they require monitoring from professionals, and they must be worn. A solution to this is to use three independent tri-axial accelerometers to send data wirelessly to be processed by a laptop. This allows for the amount of deflection to be determined when a person's weight is applied to the mattress. If deflection exceeds a predetermined threshold then movement has occurred. An advantage of this approach is that the true orientation of the accelerometer does not have to be determined in order to measure movement. Monitoring can also continue even if the sensors have been knocked out of place. Other advantages to this method are that noise is eliminated, and there is increased scalability and lower storage requirements due to the fact only discrete events are stored [44].

Table 2-4: Internet of Things Enabled Technology for Monitoring Depression in Elderly People

Sensors	Algorithms	Applications	Advantages	Disadvantages
Sleep Monitoring [44]	Formant Tracking Algorithm [45]	Touchscreen driven UI [44]	Amount of sleep can signal whether someone is depressed	Wearing the sensor overnight could be irritating for the person.
Video Monitoring [45]	Dictionary Learning Algorithm [45]	Daybuilder [46]	Video monitoring is accurate in detecting signs of depression.	Privacy concerns can outweigh the benefits from using video monitoring.
Audio Monitoring [45], [47]	Minimal-redundancy-maximal-relevance [47]	BBD Android Application [48]	Voices of depressed persons have specific characteristics that allow for detection of symptoms. Also non-invasive and portable.	If sensor is forgotten then data collection cannot be performed.
Body Sensors [43]	Machine Learning Algorithms [43]	Mobilyze [43]	A sensor network can perform constant monitoring of activities.	Relies on the person to remember to wear the sensors. Persons could be irritated by wearing devices constantly.

2.2.5. Sensor Fusion

A significant number of studies have been carried out into multiple sensor based activity recognition [49]. Results, typically, have a high accuracy and there is a low computational load on each sensor. However, the battery consumption of the devices is high when communicating wirelessly. The drawbacks of using a single accelerometer are due to the inability to distinguish an activity as mentioned by [50]. This is addressed by placing multiple accelerometer devices across the participant's body, or by combining accelerometers with other sensor types such as GPS in an attempt to improve accuracy. Classifying physical activity using features extracted multiple sensors or a network of accelerometers have typically made use of the K-nearest neighbour (KNN) and naïve Bayes (NB) techniques.

Using an SVM algorithm to fuse data collected from various sensors is investigated by [50] in order to more accurately determine the physical activity. This is done using SVM as it can calculate a decision boundary to separate activities from one another. For multiple activities, they take a "one against one" approach to separate them and produce a model for each. Each model produced will be tested against a data point, which will then receive a vote to decide which activity should be associated to it. The activity with the majority of votes will be identified as the new data point that the activity is associated with.

A system based on a network of multiple wireless-interconnected-medical sensors is proposed by [51]. This set-up allows for the collection of medical data from typical daily activities. They note that the typical solution of a single versatile system is less flexible and takes longer to design and implement. Instead, the multi-sensor solution provides the benefit of the components being ready to use.

Bluetooth was the wireless system used as is the standard [51] for ACOR+ kinematic system. This system combines a tri-axis accelerometer, a microcontroller for processing signals and Bluetooth for wireless communication. The device can be worn on the hip during the day using a belt clip, or on an adjustable chest belt for night-time monitoring. Participants wore the ACOR+ on their hips for 8 hours during the daytime to record their activities.

Participants were fitted with an ActiGraph uniaxial accelerometer and a custom designed activity monitor based on the IDEEA monitor for a study by [52]. The custom activity monitor consisted of an array of five accelerometers attached to the skin in multiple locations on the body and connected to a hip pack for data for recording. An ANN approach was used to create the model from raw acceleration features. Around 112,000 minutes of data were used to train the model, collected from 102 subjects. Only 81 subjects produced data from both the IDEEA and ActiGraph devices.

Four ShimmerTM wireless sensors were used in [53] study. Each sensor was attached to the chest, left under-arm, thigh and waist on each of the eight participants. The eight participants were asked to perform a series of eight activities in the home which are identified by [53] and were repeated three times. The tri-axial accelerometer data had a sample rate of 200 Hz and a 12-bit resolution. Data was transmitted via Bluetooth through a wireless body area network allowing the four sensors to transmit simultaneously. Five classifiers, ANN, KNN, Naïve Bayes, Decision trees and SVM were compared using the data set produced from the accelerometers. The accuracy of the classifiers was evaluated using a 10-fold cross-validation.

Data from twenty participants was used in an experiment by [54]. Data was collected from accelerometers located on the hip and wrist of the participants while they performed a series of typical daily activities. The sampling rate of the devices was set to 76.25Hz and data was collected for 54-131 minutes under experimental conditions and 82-160 minutes under semi-naturalized conditions. A leave-one-subject-out cross-validation approach was used during the classification of features extracted from the raw accelerometer data [54]. Nineteen data sets were used in the training of the data set, with the process repeated for all twenty subjects. Activity recognition is achieved using just the sensor on the hip and wrist, fusion of the features from the two sensors and finally from using a Naïve Bayes classifier to fuse the classification results.

2.3. Processing Layer

Signal information received from the sensors is stored and analysed in the processing layer. Here the data is pre-processed before feature extraction can be used. The resulting features are then classified in order to determine PA.

2.3.1. Data Pre-Processing

Typically, in PAR, time-series segmentation methods are used in order to match PA patterns obtained from multiple sensors on the body or in the environment, which are consecutively activated. Data sets produced by the sensors are broken down with temporal series using time windows. There are two common algorithms used for breaking down PA, they are the sliding window algorithm and the Sliding Window and Bottom-up algorithm [55]. With Sliding Window, the left point of a potential segment is anchored at the first data point in the time-series. Using increasingly longer time segments, it then attempts to approximate the data to the right of the anchor. The algorithm is simple and intuitive and is also an online algorithm, which is the main reason why it appeals to the medical community for patient monitoring.

The sliding window algorithm, is mostly employed using fixed temporal segments that can be over-lapping and non-overlapping [56]–[60] and can be seen in Figure 2-3 below. Non-overlapping instances of time windows are comprised of varied lengths of input [61], [62]. This splits the activity up with continuous signals from sensors and can produce the wrong output as a result. Overlapping instances of sliding window with 50% [59], 75% and 90% overlapping [63] tend to produce a higher accuracy for PAR as the amount of overlap increases, however this can be more resource intensive.

Dynamic segmentation of activities does not use a fixed length for the time window. Instead activities are extracted from the data when they are detected during evaluation [64]–[67]. This approach allows for the continuous segmentation and aggregation of data on a timeline allowing for continuous on-going PAR. User specified thresholds [65] and heuristic probability approaches [66] are typically used for dynamic segmentation of time windows. Sliding window and bottom up algorithm is an approach that combines the two algorithms to produce higher accuracies in results; however, it is a more complicated approach [55]. It is an online approach due to the inclusion of the sliding window algorithm and allows for the global view on the data collected. The method has successfully been applied to gesture identification and activity recognition studies using accelerometers and gyroscopes [68]–[70].

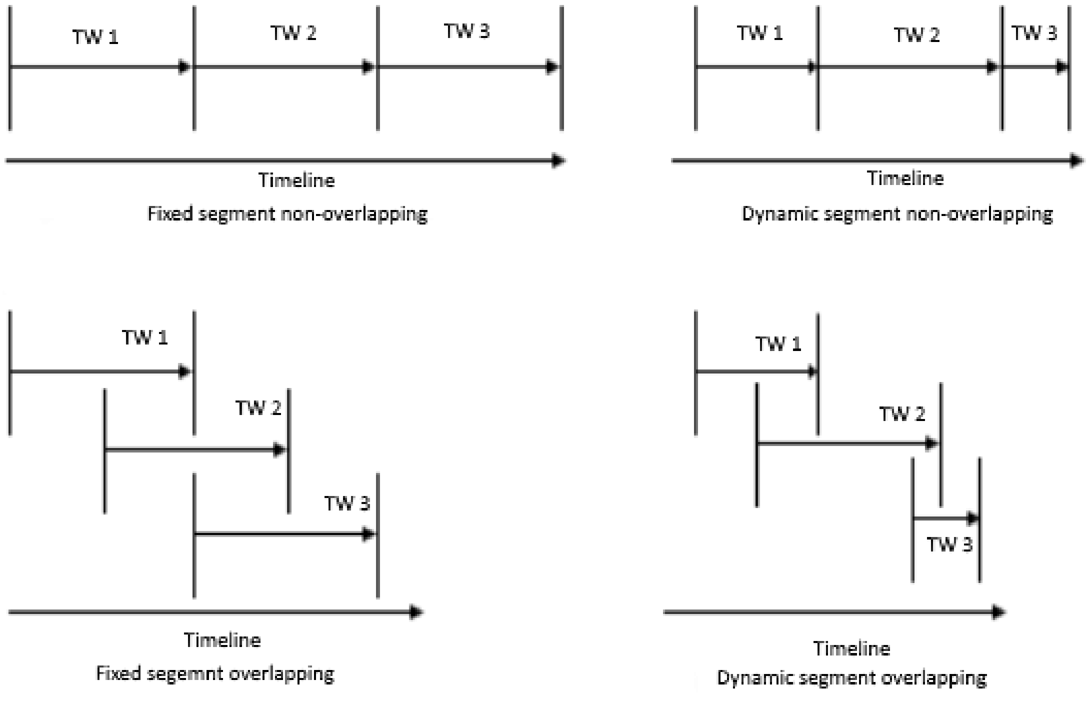

TW 1
TW 2
TW 3
Timeline
Fixed segment non-overlapping
TW 1
TW 2
TW 3
Timeline
Dynamic segment non-overlapping
TW 1
TW 2
TW 3
Timeline
Fixed segemnt overlapping
TW 1
TW 2
TW 3
Timeline
Dynamic segment overlapping

Figure 2-3: Time window segmentation

Many of the existing studies in literature display a high accuracy for recognition using online methods of time segmentation. The issue many studies attempt to deal with is the window sizes, which can be based on the signals or the environment. Misclassification of activities can occur on short (2s – 6.7s) and long-time (10s – 12.8s) static windows, especially on transitional activities [64]. This may be due to transitional activities taking longer to complete than dynamic or static activities. Long-time windows can also lead to misclassification due to multiple activity signals and redundant information. Shorter windows can lead to key information being lost resulting in an inaccurate classification [59]. However, many studies that incorporate a static sliding window do achieve a high degree of accuracy, while Dynamic sliding window techniques do improve accuracy but come at an increase in computational complexity [59].

2.3.2. Feature Extraction

Feature extraction is the process of extracting useful features from a data set in order for them to be classified. Signals from sensors are analysed in order to remove redundancy and produce distinguishable features that can be used to classify activities. Four types of features have been categorised and used throughout the literature, these can be found in Table 2-5.

Table 2-5: Feature types that can be applied to PAR.

Feature Type	**Extracted Features**
Time Domain	Mid-value, Minimum, Maximum, Standard Deviation, Average, Range, Variance and Root-mean square, Correlation, Covariance
Frequency Domain	Spectral Energy, Entropy, Coefficients, DC Component, Magnitude, Peak Amplitude, Skewness, Kurtosis
Biometric Domain	Cepstral features, R-Peaks
Other	Linear Discriminant Analysis, Principle Component Analysis, Dynamic Time Warping

Time domain features are statistical metrics that occur randomly in constant signal changes over time. Typical features of time domain are mid-value, minimum, maximum, standard deviation, average, range, variance and root-mean square [71]. Calculating the average can be used to smooth any outlying data from a signal. Variance describes the distance from the expected mean value [72]. Standard Deviation is used to determine the spread of the data points from the average value [73]. Root mean square is a quadratic classification that is typically the most significant parameter of a wavelet function that can be used for extracting features of dynamic activities [74]. Rybina et al. [75] used correlation coefficient to identify the activities; hopping, running, jumping, balancing and skipping which were collected using four accelerometers attach to the arms and thighs. Wang et al. [71] uses time-domain features on accelerometer data to identify: staying still, walking, running, going upstairs and going downstairs. A tri-axial accelerometer embedded in a smart phone was used to collect the data. Gan [76] places a smart phone on the participants arm, containing a tri-axial accelerometer. They perform a hammering activity with three different degrees of force. Mean and standard deviation are applied to the signals generated. Time-domain features are easy to process and can be time saving however, they cannot describe states of motion accurately [77].

Frequency domain features are typically extracted using Fast Fourier Transform (FFT) by transforming the time domain signals into the frequency domain [78]. An FFT output creates a set of base coefficients in order to represent the amplitudes of the frequency component of the signal and its distributed energy. Spectral Energy, Entropy [79], Coefficients [78], DC Component [80], Magnitude [81], Peak Ampltiude [76], Skewness and Kurtosis [72], [82] are examples of frequency domain features. Spectral energy is the sum of the FFT coefficients. Entropy is calculated using normalized entropy of FFT magnitudes of the signal in order to distinguish different activities with similar energy values

[79]. DC Component is the average acceleration value of the input signal over the window [81]. These features have been linked to activities such as walking and running. Only the frequency content of a signal can be extracted using FFT analysis. However, wavelet analysis can be used to extract time and frequency-based features from a signal. It can be formulated using a continuous or discrete wavelet transform [77], [78]. Discrete wavelet transform uses a set of base functions which decompose a signal into detailed and approximate signals based on the original [83]. A continuous wavelet transform provides a signal decomposition at different scales [84]. Walking, jogging, running, standing are examples of activities that have had wavelet transformation applied to them [77], [83].

When it comes to biometric signals, methods of feature extraction applied to the time and frequency domains are limited in their capabilities. One solution is the use of Cepstral domain features which can be used to analyses ECG signals for PA detection [85] or for recognising speech information [86]. Calculating Cepstral domain features involves using short fixed length windows for processing, and as a result it does not require a pre-processing step for segmenting and normalizing heartbeat signals [85]. R-Peak detection is another feature that can be used on ECG signals [87], [88]. R-Peaks involve detecting the peak of the R-wave, which represents the ventricular depolarisation of a heartbeat, the most distinguishable feature of the ECG signal. The Pan-Tompkins method can be used in R-Peak detection as demonstrated by [87].

Linear discriminant analysis (LDA) which is a process that involves reducing feature dimensions in order to reduce computational complexity [88] and for classification purposes [89]. The optimal discrimination matrix is chosen from the ratio maximization of the determinant of the between and within-class scatter matrices [90]. LDA can be applied to transitional activities, walking, running and static activities. Principal component analysis (PCA) can be used to identify a smaller number of uncorrelated variables within a large data set [91]. This results in the reduction of the data set size without any significant loss of information. The average, variance and covariance of the data set are required in order for PCA to work. Lahiri et al. [92] applies PCA to the data set obtained from using a Microsoft Kinect sensor. This recorded chest pain, fainting and headache related activities. Dynamic time warping (DTW) is an approach that recursively projects a warp path from a lower resolution to a higher resolution and refines it at each step until the optimal match is found [93]. The three main steps defined, involved shrinking the time series in smaller steps in order to represent the same curve with optimal accuracy and fewer data points, projecting a warp path from a lower resolution to a higher one and finally refining the warp path projected. DTW has been applied to several human activities such as waving, clapping, walking [93], rowing, boxing, sitting [94].

Many studies extract time-domain features from their signals, as the data is easy to process, can be time saving and can be extracted in real time. However, time domain features alone can't describe states of motion accurately [77]. Frequency domain features can produce a higher accuracy compared to time domain features, however, this leads to an increase in computation in order to differentiate between different activities [83]. As a result, studies have taken to using wavelet models as the incorporate both time and frequency domain features in order to improve accuracy [83], [84].

2.3.3. Classification

Classification and clustering are essential aspects of machine learning and vital for the classification of activities. Supervised learning techniques, as outlined in Table 2-6 appear frequently in previous studies on PAR. Unsupervised and semi-supervised techniques have also been adopted for some studies. Supervised learning techniques train their models on data sets that have been labelled with the desired outcome information, i.e. the activities we are trying to recognise. Unsupervised techniques do not contain labels in the data set while semi-supervised contains mostly unlabelled data but can include some labelled.

Table 2-6: Supervised learning techniques used in AR

Classifier	Reference	Sensors Used	Activities	Accuracy
ANN	[95]	Video Recording	Walk, run, stumble, limp, forward, backward and sideways falls, bend, sit and lie	83.33 – 96.66%
	[96]	3D Accelerometer and Gyroscopes	Five data sets containing, walking, walking up and down stairs, sitting, standing etc.	96 – 100%, 94 – 100%, 65 – 97%, 83 – 96 %, 60 – 97%
	[97]	Video Recording	Shop enter, exit, re-enter, window shopping, browsing, immobile and walking	60 – 98%
	[98]	Single-channel electrodes	Pinch, finger spread, fist, wave in, wave out, relax	89.4 – 96.4%
	[99]	Video Recording	Walking, Running, Jumping, Tripping	92 – 100%
	[100]	77 Simple and Ubiquitous Sensors in an apartment	Bathing, cleaning, laundry, dressing, going to work, grooming, preparing a: beverage, snack, breakfast, lunch, dinner, going to the toilet, washing dishes	1.02 – 19.98% 2.02 – 20.2% 1.01 – 19.96%
DT	[101]	iGwatch	Standing, walking, running, walking upstairs and down-stairs	83.6 – 98.4%
	[102]	Nokia Motion Wristbands and PDA	Walking, lying, sitting, standing, bicycling and running	79 – 100%

	[103]	Tri-axial accelerometer	Standing, lying, walking, sitting and dining, sitting and non-dining	81.1%
	[104]	Tri-axial accelerometer	15 activities split into 5 categories including walking, running, sitting and lying	84 – 100%
SVM	[105]	Actimeters, Microphones, PIR, Door contacts and environmental sensors	Sleeping, resting, dressing, eating, toilet use, hygiene and communication	80 – 97%
	[106]	Video Capture	Walk, Run, Stumble, Limp, Forward Fall, Backward Fall, Sideways Fall, Sit, Lie	83.33 – 95%
	[107]	Inertial Sensors & Barometer	Walking, Upstairs, Downstairs, fall, run, lie, sit, stand	96 – 100%
	[108]	Smartphone sensors	Sitting, standing, walking, upstairs, downstairs, lying	56.97 – 100%
Naïve Bayes	[109]	Infrared Camera	Walking, running forwards/backwards, carry gun, carry backpack, dropping, searching, digging.	76%
	[110]	EEG electrode position system	Emotions happy and unhappy	36 – 75%
	[111]	Tri-axial accelerometer	Sitting, Standing, Lying, Walking, Walking upstairs	36.1 – 100%, 60 – 100%, 53.13 – 100%, 57.14 – 100%, 61.7 – 100%
	[112]	Tri-axial accelerometer	Walking, jumping, running, sit-to-stand/stand-to-sit, stand-to-kneel-to-stand, standing	95-6 – 99.1%
HMM	[113]	State Sensors	Bathing, toileting, going to work, preparing lunch, preparing dinner, breakfast, dressing, grooming, preparing a snack, beverage, washing dishes, doing laundry, cleaning, washing hands, putting away dishes/groceries/laundry, watching tv, going out, lawn work.	0 – 100%
	[114]	Video Monitoring	Rush, Carry, Bend, Walk	85 – 95%
	[115]	Stereo Camera	Left-hand up-down, right-hand up-down, both hands up-down, boxing, left-leg up-down, right-leg up-down	87.5 – 95%
	[116]	Kinect Sensor	Stand, sit, drink, talk, read, stretch, akimbo, come, follow, stop, walk	88 – 100%

a) Artificial Neural Network

Artificial Neural Networks (ANN) are a method of classification that can be found in many studies for activity classification [117]. They are data driven methods, that are also self-adaptive, meaning they can adjust to the data without any specified functional or distribution form in the underlying model [118]. ANNs are designed to simulate the biological neural networks in the brain and how they process information. ANNs learn from the patterns and relationships they detects in a data set and can produce high accuracy from large data sets. The network, as shown in Figure 2-4 is made up of neurons, which transmit a signal to a connecting neuron for processing. The connections and the neurons typically have a weight associated to them that increases or decreases the signal strength at the connection as the network continues to learn. An ANN is comprised of multiple layers, the first being the input layer and the final layer being the output. Signals may traverse the hidden layer in between those two points where data transformation may occur on the inputs. Appendix A displays a simple prototype neural network that contains the back-propagation, feedforward calculations and weight adjustments in order to aid in understanding how a neural network functions.

Within the hidden layer, the summation of the inputs and the weight are added to the bias. The bias is an additional weight, but it is also a constant, which exists to help the model to provide the best fit of the provided data. The summation of inputs and the bias can be defined below where x are the inputs to the neuron, w are the weights, n are the number of inputs and i is a counter from 0 to n.

$$\sum_{i=1}^{n}(w_i\, x_i) + bias$$

Equation 2-1: Neuron calculation for ANN.

ReLu activation functions can be used to calculate a neurons decision, and is the most common activation function to be used in classification models. The function will return the value x for any positive value it calculates, and will return 0 if it receives any negative input.

$$f(x) = \max(0, x)$$

Equation 2-2: ReLu Activation Function

Backpropagation is a method used for updating the weights using gradient decent. The partial derivative of the error with respect to w_i is subtracted, with a representing the learning rate. The calculation proceeds backwards throughout the network, and the derivation of error is evaluated by applying the chain rule.

$$w_i = w_i - a(\frac{\partial E}{\partial w_i})$$

Equation 2-3: Calcuation for Backpropogation of Weights

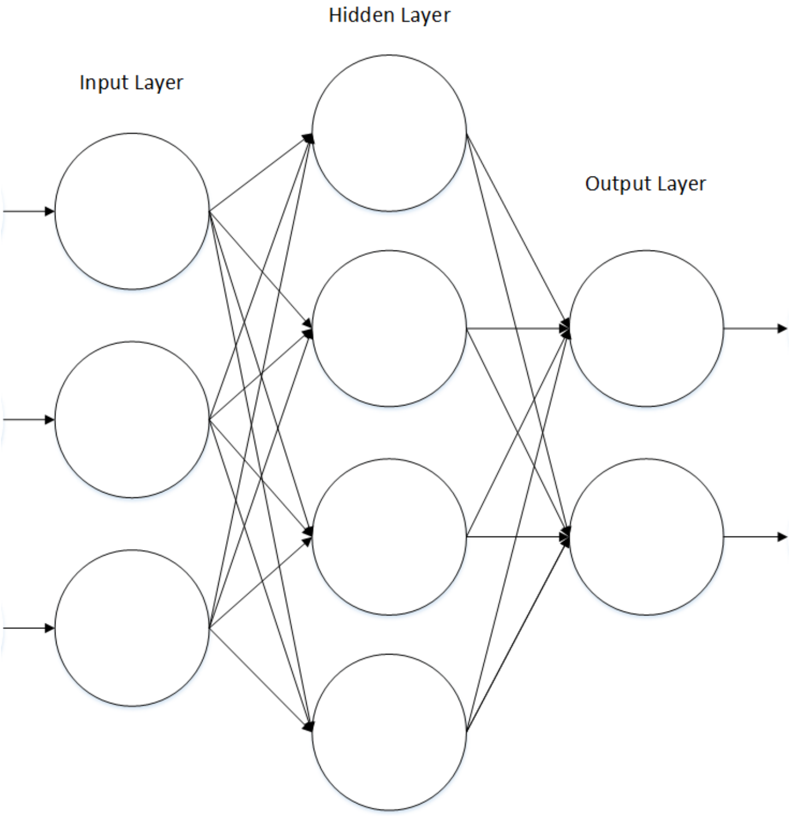


Figure 2-4: Example of an ANN feedforward network

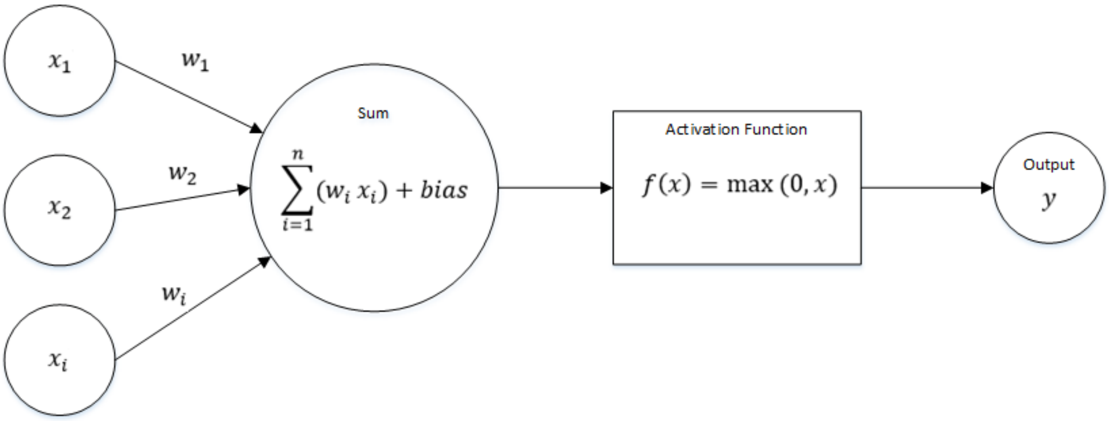


Figure 2-5: ANN Neuron

A fall detection system that attempts to extract features from video sequences was proposed by [95]. 24 participants were used to perform 10 activities such as walking, running and a variety of falls. Video recordings of each activity were collected in order for their neural network to be applied for classification. Accuracy of the classification ranged from 83.33 – 96.66%. Neural networks were applied to 5 different data sets that were acquired by [96] that collected data of various activities in different environments. Sensors included 3D accelerometers and gyroscopes. Activities included walking upstairs and downstairs, running, jumping and standing. Their neural network achieved various degrees of accuracy on each data set, with one set achieving 94 – 100% and another achieving 65 – 97%.

b) Decision Trees

Decision Trees (DT) are trees where the inputs from data sets can be used for tests on the internal nodes and where the leaf nodes will be considered categories for the data sets. Each test is filtered down through the tree in order to achieve the desired output of the input set [119]. DTs aim to break up complex decisions into multiple smaller, simpler problems through a multi-stage approach. They consist of nodes that are formed from the "root" of a tree. The root being a node that has no input edges [120]. Nodes that have outgoing edges are known as test nodes, with the decision nodes being known as leaf nodes. All nodes have exactly one incoming edge, represented by an arrow in Figure 2-6 below. The leaf node is assigned to one class, and represents the target value; however, it may also hold a probability vector in order to indicate the probability of a target value. The test nodes split the DT instance into two or more subspaces according to the values from the input. Each subspace has a condition associated with it that must be met in order to classify the incoming data, for numerical values, the condition represents the range of input values. The DT instance is classified by navigating the tree from the root down to the appropriate leaf depending on the results from each of the test nodes and conditions.

DTs are attractive compared to single-stage classifiers as a tree classifier can reduce unnecessary computations by only testing a sample against certain class subsets [121]. They are also fairly flexible and have the ability to choose different feature subsets at different internal nodes of the tree in order for the discrimination of feature subsets to be performed optimally, potentially providing a performance increase. However, designing an optimal tree may prove difficult and this can affect the overall performance of the classifier. There is a risk of overlap, particularly when there is a high number of classes in the tree. Errors also have the potential to accumulate as progression down the tree is made.

We calculate entropy, which is how a decision tree decides how to split its data, thus drawing the boundaries of the tree. S represents the boundaries of the tree and p_i is the probability of an element of the i-th value in the data set.

$$E(S) = \sum_{i=1}^{c} - p_i \; log_2 \, p_i$$

Equation 2-4: Entropy calcuation for DT

From the entropy, we can calculate the information gain represented by $IG(S, A)$. This is a measure of difference in entropy from before the data set S is split on the attribute A.

$$IG(S, A) = E(S) - \sum_{i \in A} P(i)E(A) = E(S) - E(S|A)$$

Equation 2-5: Information Gain calculation for DT

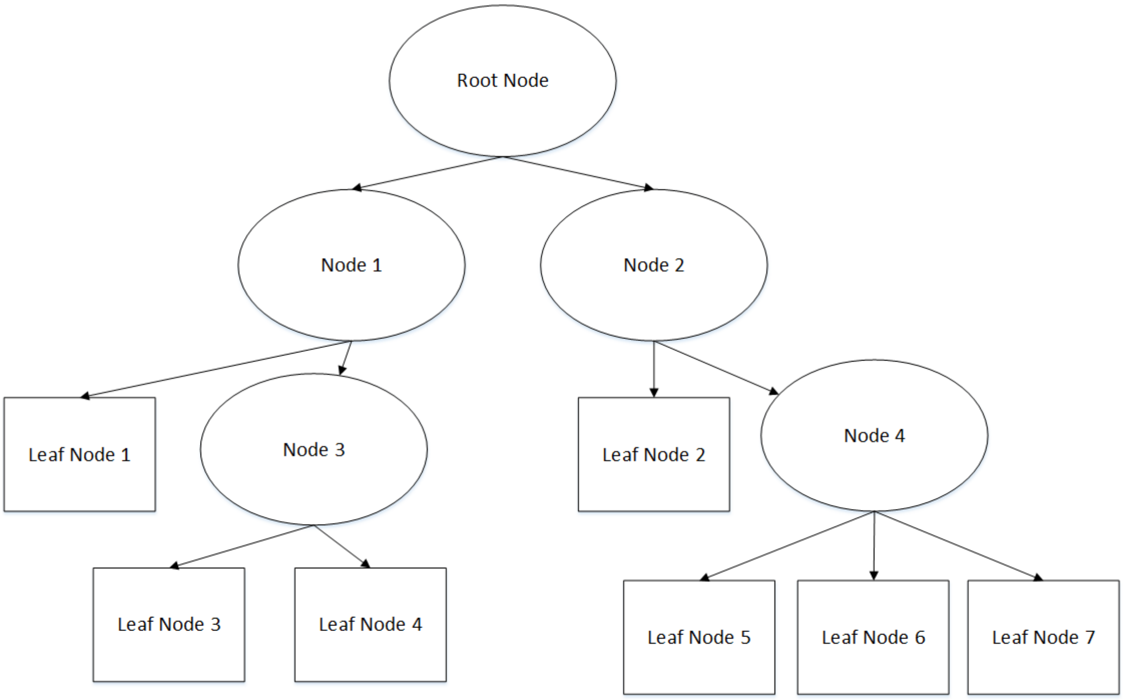


Figure 2-6: A Simple Decision Tree Example

An activity recognition method using data collected from a smart watch is presented by [101]. A CART decision tree is used for behaviour classification, with the output decisions being either true or false. An iGwatch is used to collect data from ten participants. Participants performed standing, walking, running, walking upstairs and down stairs activities achieving an accuracy of 83.6 – 98.4%. An automatic activity recognition system using a PDA and wireless motion bands is evaluated by [102]. The Nokia wireless motion bands contained 3D accelerometers and transmitted the activity data via Bluetooth to the PDA for classification. Seven participants performed walking, lying, sitting, standing, bicycling and running activities. Collected data is fed into one of four nodes of the decision tree classifier. The DT classifier achieved accuracies of 79 – 100%. Tri-axial accelerometers were worn on the right and left wrist and on the waist in order to collect motion data in the study by [103]. Participants performed standing, lying, walking, sitting and dining, sitting and non-dining for various degrees of time. A 5-second window is applied for feature extraction, with five features being extracted from data. DT was implemented to classify the activity data from all three sensors achieving an accuracy of 81.1%. Fifteen participants using three tri-axial accelerometers performed eighteen activities for [104]. Activities included walking, sitting, lying and running and were performed for 45 seconds each and repeated twice. Mean, energy, entropy and correlation features were extracted from the acceleration data. Activities were split into five categories before classification using DT, which involved a leave-one-subject-out approach. Accuracy rates achieve range from 84 – 100%.

c) Support Vector Machine

A *Support Vector Machine (SVM)* is a supervised learning model which, takes advantage of the plane concept to describe decision boundaries [122]. The decision plane separates a series of objects that belong to different classes. Classification is achieved through the creation of hyper-planes in multi-dimensional spaces. SVM can support categorical and continuous variables for classification and regression. Categorical variables are determined using a dummy variable, which is set to either 0 or 1. The SVM algorithm then develops the optimal hyper-plane to reduce the error function.

SVMs are a type of pattern classifier that minimizes the empirical training error and are based upon the structural risk management principle [123] . They have demonstrated an ability to generalize effectively when dealing with small samples of training data compared to ANN. A hyperplane, as shown in Figure 2-7, is constructed by SVM which is used to classify linearly separable patterns [124]. They aim to minimize the upper bound of the generalization error by maximizing the margin between the data at the hyperplane that separates it in order to correctly classify a given pattern. The larger the margin size, the more accurate the classification of the pattern. The classifier complexity and error can be explicitly controlled allowing the classifier to scale well to high dimensional data. However, SVM requires a good kernel function in order to operate optimally.

The hyperplane of a linear SVM can be defined as follows, where w is the normal vector to the hyperplane, with x representing the set of points.

$$w \cdot x + b = 0$$

Equation 2-6: SVM Hyperplane calculation.

The decision function takes the data set as an input and gives the decision as on output, it can be defined as:

$$f(x) = sign((w \cdot x) + b)$$

Equation 2-7: SVM decision function.

The functional margin gives the position of the data point in respect to the plane. y_i represents the i-th label of the training example.

$$\hat{\gamma} = y_i(w^T x + b)$$

Equation 2-8: SVM functonal margin calcualtion.

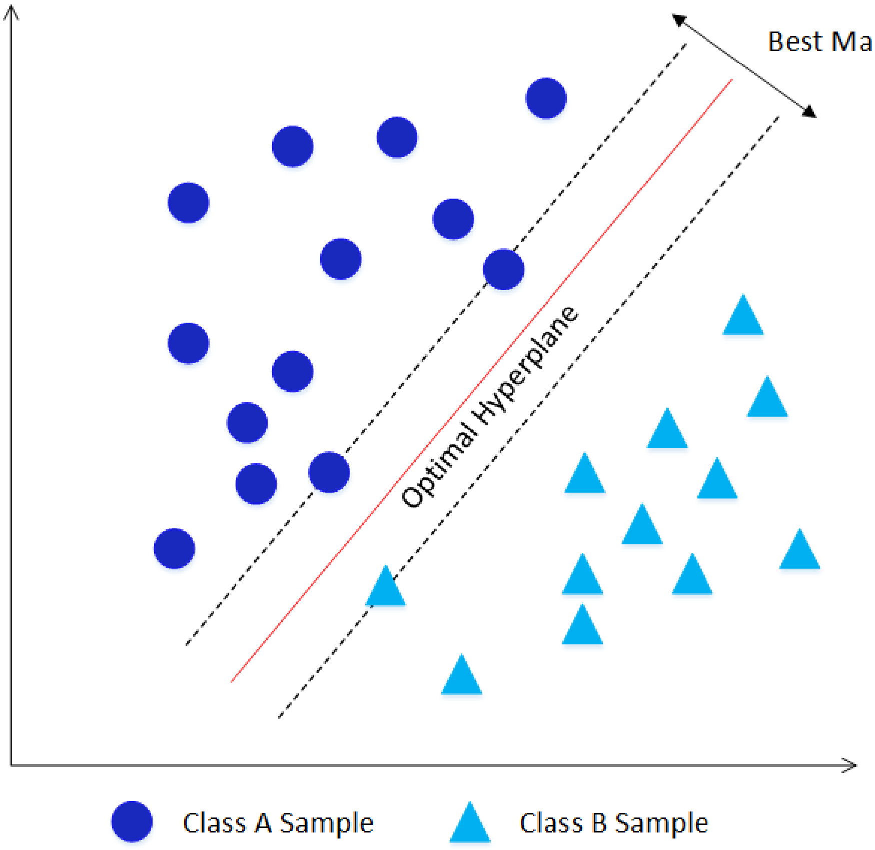


Figure 2-7: SVM example displaying the hyperplane and margin

A study by [105] uses SVM on sensor data collect from ADL. Sensors consisted of Actimeters, Microphones, PIR, Door contacts and environmental sensors placed around the home. These sensors monitored sleeping, resting, dressing, eating, toilet use, hygiene and communication activities performed by thirteen participants. SVM was used to classify the activities achieving accuracies of 80 – 97%. A robust fall detection system that uses SVM for classification of activities is proposed by [106]. Video recordings in a controlled lab were performed on daily activities and various types of fall in order to train the algorithm. Forty-eight participants of different heights, weights and genders performed the various activities and repeated them five times. A One-Against-One method of SVM classification is used to identify the activities. Accuracy rates of classification ranged from 83.33 – 95%. Inertial sensors and barometers were used to collect activity data in a study by [107]. The sensors were placed on the waist of six participants in order for them to perform eight ADL. These included, walking, running and sitting. SVM is used for classification with a 10-fold cross-validation approach being applied during the training phase. Classification managed to achieve 96 – 100% accuracy. Integrated sensors in an android smartphone were used for AR in the study by [108]. Ten participants performed six activities including, walking, sitting and lying. SVM was applied in order to classify the activity data achieving accuracies of 56.97 – 100%. This was due to the lying activity being classified as sitting.

Naïve Bayes (NB) is a simple probabilistic classifier assuming there is a strong independence between the features within a data set [125]. Even as a simple classifier, it is capable of outperforming solutions that are more complex. There are several variations of NB that can be deployed; Multinomial – which is mostly used to document classification problems, Bernoulli – which is similar to multinomial but the predictors are Booleans and Gaussian – where predictors are continuous values and are not discrete. NB is a suitable classifier for multi-sensor fusion assuming that all the features are independent from each other. Multi-sensor fusion can be represented by the probabilities of each sensor.

Thermal images of outdoor human activities are captured using an infrared camera in a study by [109]. Eight participants performed a series of activities including walking, running; crawling and further activities were provided in a data set. NB is used to classify feature vectors achieving a 90% recognition rate on recorded infrared video. In total, recognition was 76% when the additional data set was included. An emotion recognition system using EEG signals is proposed by [110]. Twelve pleasant and unpleasant images where shown to 26 participants. The EEG electrodes were placed on various points on the head for data collection. NB was one of several classification methods used and produced an accuracy of 36 – 75%. A tri-axial accelerometer is employed by [111] for AR using an NB classifier. NB was chosen, as it was easy to implement and produced a model fairly quickly. One participant performed sitting, standing, lying, walking and walking upstairs activities. Activities were repeated and recorded at different sample rates for comparison. Overall accuracies ranged from 84 – 87.8%. Two participants performed six activities while wearing tri-axial accelerometers in a study by [112]. All three accelerometers were placed on the waist at various positions. Activities included, jumping, running and walking and were classified using NB. Accuracies ranged from 95.6 – 99.1%.

A *Hidden Markov Model (HMM)* is a classification method that is defined by a set of states with interconnections between them. A set of prior probabilities for input and emission probabilities for the output of the states and a set of transition probabilities are typically defined for HMM [113]. A reconfigurable HMM is presented by [113] for AR. Data was collected in a lab using 77 on/off states sensors. Sensors are activated and deactivated when a participant is performing an ADL. Activities included bathing, going to the toilet, washing hands and cleaning with accuracies ranging from 0 – 100%. Thirty participants were recording performed natural activities using a stationary camera [114]. Activities included walking, rushing, carrying and bending. For each activity, the participant entered the monitoring domain, performed the activity then left the domain. Recognition was trained using HMM producing high accuracies of 85 – 95%. A stereo camera is used to recognise activity by monitoring joint angle in a system proposed by [115]. Estimated joint angle features are taken from time-based activity video frames and mapped in order to create discrete symbols for HMM. Participants performed boxing, moving both hands up and down as well as individually, and moving each leg up and down. Results were compared against a binary silhouette-based solution. The joint angle based system outperformed the silhouette solution achieving accuracies of 87.5 – 95%. A Kinect sensor is used to capture 3-D skeleton joints in a system proposed by [116] and treats the collect activity data as a time series of representative 3D poses. RGB images, depth and skeletal information is provided by the Kinect at 30fps with a depth of 4 to 11 feet. Activities included sitting, standing, drinking, stretching and reading and were performed by five participants, five times. HMM was applied to the data for classification achieving high accuracies of 88 – 100%.

K-means clustering is a partitioning based, non-hierarchical clustering method and a form of an unsupervised learning algorithm [59]. It begins by initializing cluster centres according to the distance between input feature vectors and cluster centres that already exist. Each centre cluster is updated by calculating the centroid of each cluster. The update occurs due to changes in each cluster and is repeated until the value of the centre cluster no longer changes. In the study by [59], K-means clustering is applied to classify fragments of segmented raw sensor data. However, while simple to implement [126] notes that K-means does not always find the global optimal solution based on the objective function and it can also be sensitive to the initial randomly selected cluster centres. It can also have issues dealing with clusters that differ in size, shape and density. As a result, for [126] study, a Gaussian Mixture Model (GMM) is deployed. GMMs optimize the fit between data and a parametric distribution. Activities in their study involved walking and running at various speeds and the GMM clustering applied to the data managed to obtain and 94.3%.

Semi-supervised learning techniques involve the use of labelled and unlabelled data for training. Labelled data requires a human to perform the labelling, which can be expensive, and time consuming and the lack of variety in the data can lead to a reduction in recognition performance. However, unlabelled data is fairly inexpensive to obtain and the combination of the two can improve the accuracy of learning [127]. Disagreement based semi-supervised learning techniques are discussed by [128]. This involves generating multiple classifiers and then exploiting the disagreement between them. The classifiers are forced to cooperate to utilise unlabelled data. A semi-supervised convolutional neural network (CNN) is presented by [129]. CNNs provide stable latent representations of each level of the network. The algorithm is also able to effectively identify the salient patterns of the activity signals. For their solution, three encoding paths are used to the labelled and unlabelled data; clean encoding, noisy encoding and decoding. During noisy encoding, labelled data has predictions applied to it using a top-level softmax classifier using cross entropy cost, while unlabelled data is decoded and then reconstructed to be the same as the corresponding clean input. Three data sets are used and have supervised and semi-supervised algorithms applied to them and are then compared. Supervised CNN achieved 48.68 – 59.73%, while semi supervised achieved 54.90 – 69.38%.

2.3.4. Discussion

Supervised learning is a well-documented approach to PAR in the literature, with many studies incorporating various algorithms into their approaches and achieving relatively high accuracies in their results. The classification methods can be simple to implement, however large data sets are sometimes required in order for the classification techniques to be successfully trained. For example neural networks may have difficulty with a weak sample of data however, SVM can deal with small data sets fairly well [105].

Data sets also require labelling in order for activities to be successfully classified. This is a process that can take a significant amount of time to do if the sample size is large enough. Humans also perform an activity in various ways, for example – people walk at different speeds, and therefore it can be difficult to correctly classify activities as a fast walk could be misinterpreted as running.

There are several studies documenting unsupervised and semi-supervised methods and their application towards activity recognition. Unsupervised techniques can be computationally efficient as they use clustering to generate the activity model. However, unsupervised techniques require a large data set in order for the underlying patterns to be discovered. If the data set is weak, then the patterns will be difficult to discover without additional supervision [128]. Semi-supervised methods take advantage of labelled and unlabelled data in order to produce better predictions of activities as the methods attempt to address the challenges posed by supervised and unsupervised methods. If labelled data is limited, performance can be improved by adding unlabelled data. Many studies into semi-supervised methods have made the assumption that feature engineering has already been performed [129]. As literature on unsupervised and semi-supervised methods is limited, more research needs to be performed on these methods, particularly on long-term activity monitoring in order to improve the accuracy of the techniques so they can compete with supervised learning methods.

2.4. Application Layer

PAM can be applied to many areas of health care (such as elderly people monitoring) through the use of a variety of sensors (ambient sensors, wearable devices, smart phones). This section will look at some of the applications of sensing technology for lifelogging, assisted living and the use of abnormality alerts for medical interventions.

2.4.1. Lifelogging

Lifelogging is the process of applying digital devices such as wearable sensors in order to capture the events of an individual's life for their health and wellbeing [9]. Initial lifelogging attempts focused on the process of image capturing via a camera for PAR [11]. Advances in technology has allowed lifelogging to be applied into more situations. Wearable devices such as smart phones and wrist sensors can be used for the continuous monitoring of an individual's PA. The use of these devices for lifelogging the intensity of PA data can typically be placed into five categories; Sedentary, Light, Moderate, Vigorous and High Intensity [130]. These categories are based on metabolic equivalents (METs) cut offs and have formed a standard of PA levels to be used in maintaining a healthy lifestyle. However, this classification offers a general instant categorization measure that produces a deficit when evaluating and assessing lifelogging PA patterns that have been accumulated.

Over recent years, commercial wearable devices and mobile applications have seen an increase in the market. Many of these support the long term recording and collection of personal health information and PA [19]. Wearable devices such as Fitbit Flex , Nike+ Fuelband, Endomondo [18], [19] are all devices that can monitor health factors such as step count, calories burnt and distance walked. These devices communicate with a mobile device via Bluetooth that is running the corresponding mobile application. However, in regards to tracking personal PA, the prior devices are unreliable due to a combination of diversity in activity patterns and various environmental factors.

Mobile applications used for lifelogging such as Moves [9], [18], are capable of supporting the long term recording and collecting of PA data through the tracking of 3D accelerometer data in the smart phone and GPS information allowing an application to track an individual's movements and activities. 3D Accelerometers and gyroscopes on smart phones can be used in conjunction with an application in order to detect when an individual has a fall, allowing for direct reporting to a care giver [21]. The PA data produced by these applications can find validation in healthcare cases challenging. However with the growth of wearable and mobile devices on the market, the diversity and accuracy of such devices in IoT based healthcare systems is improving [19].

2.4.2. Assisted Living

One healthcare challenge that has persisted over recent years, is the ability to ensure those who require special attention such as elderly people or those with disabilities are covered with professional care while the cost of maintaining such coverage increases. AAL systems are a measure that can be applied to healthcare in order to support those with disabilities or elderly people as they go about their daily lives [131]. This technology can be integrated within people's homes, allowing for safe lifestyle and continuous monitoring of daily activities. Previous research into this field focused on improving the quality of life of people in their own homes and supporting nurses and physicians in hospitals. However, less research has been conducted in helping support caregivers, who assist those with disabilities within an assisted living environment.

An example of a system used in a home environment for the tracking of elderly people is proposed by [132]. The system is capable of tracking the location of the individual within the home and monitoring motions that they might perform. A BLE wearable device, which contained a motion sensor and a set of BLE scanners, was used in the system. The wearable device tracked the motion type performed and broadcast the information to a server. The BLE scanners scan for the wearable device in order to determine its location based on the receiver signal strength. The data is published to an IoT platform for remote location and motion tracking. An SVM classifier was used with a 10-fold cross validation and linear kernel. This produced results of 99% accuracy for location tracking and 99.71% for motion tracking.

A common problem in existing work is that the privacy of the individual being monitored cannot be guaranteed. A typical way to help keep collected data secure is to implement secure channels for collection, transmission and storage [133]. However, this does not solve privacy issues when it comes to over-collecting information and personal data. An example of this can be the collection of video data, as many people will not want to be continuously monitored through video cameras even when told the system is secure. Any system that is put into place must also adhere to GDPR in order to preserve the privacy of those involved. A methodology to try and preserve privacy while attempting to predict human behaviour is proposed by [133]. Several sensors are placed within the sensing area and are connected to the cloud via Wi-Fi for data collection and storage. Participants performed activities such as walking or walking slowly. The aim of the study is to determine if highly private data sets can produce accurate predictions of behaviours.

2.4.3. Abnormality Alerts

Medical home alert systems were developed as early as the 1970s to assist elderly people within their own home [134]. The principle behind these systems was for a sick or elderly person to contact someone for help using a dedicated phone line. These systems have evolved over the last few decades to incorporate new sensing technology that has emerged. Ambient sensors in smart homes, wearable sensors and smart phones can be used to alert a relative or caregiver about any abnormalities that an individual may encounter such as a fall or wandering.

An accelerometer attached to a Tiva C Launchpad was implemented in [134] fall detection system. An amazon echo is used in the system to allow the user to confirm via a voice command if a detected fall is correct or false, adding a layer of confirmation. If a positive response to the fall is received, then the system distributes a text message to the primary caregiver. This text message includes a link to a live video feed in order for the caregiver to identify the state the fallen individual is in so they can call for an ambulance. If the caregiver does not respond, then the user is asked if they would like to contact emergency services directly.

The fall detection system presented by [21] uses an android smart phone application to detect falls and distinguish them from ADLs. When the application detects a fall, an alarm is triggered in order to draw the attention of nearby people. It will also send out a text message and an email to a set of contacts with details of the user's location and the time of the fall. Only the phone's tri-axial accelerometer is used to collect data in order to save power on the device. Other power optimizations include setting the sampling rate to 4Hz when the user is motionless. 28 participants were used to record 10 types of falls, 3 times each. Decision trees, K-Nearest Neighbour and Naïve Bayes classifiers were implemented, using a ten-fold cross validation strategy for evaluation. Accuracies of 92% for DT, 80% for K-NN and 89% for Naïve Bayes were recorded.

A system to alert caregivers that an individual is wandering is designed by [26]. It uses the GPS on android devices to track the user's location, and also records information such as time of day and weather conditions. This information is evaluated using Bayesian network techniques in order to determine the probability of the user wandering. Safe zones are created on the app, typically set to the patient's home. Here the user would not be considered at risk of wandering, however if they leave the safe zone, the probability of wandering increases. An SVM classifier is used to determine if the current wandering behaviour is normal or abnormal. If the user is considered to be wandering by the system, then notifications are given to the user to determine if they are okay, this allows for the reduction in false positives. If no response is given or they respond negatively, then the app will attempt to direct the user to a safe zone. Alerts are also sent to caregivers in the form of text messages and email, providing details of the individual's GPS location. Google voice is also used to connect the caregiver to the wandering individual and placing them on speed dial to establish communication.

2.4.4. Remote Health Monitoring

Remote health monitoring revolves around the concept of an operator reacting to an abnormal situation in real time without the need to be present at the situation [135]. This is appealing to researchers and clinicians as it reduces the cost of operation due to limiting the physical presence required at the site. It also provides an alternative to hospital monitoring and can facilitate the early discharge of patients who would otherwise have to remain in hospital care [136]. Early detection of deterioration can be used to identify potential hospitalisations, which can aid in reducing the number of unnecessary hospital admissions and allows health care professionals to monitor more individuals with long-term chronic illnesses. As a result, elderly patients are able to spend longer in their own homes and maintain their independence by having the monitoring performed using wearable electronic sensors, smartphones, or smart homes.

Tiny sensor nodes placed on the human body for remote monitoring are used in a remote monitoring application developed by [137]. The sensors were used for monitoring several vital signs of the patients including ECG, oxygen saturation and heart rate. Data is transmitted wirelessly which is collected by a central node. A PC control displays and records the vital signs in waveform. Sensors were tiny in order to ensure they were lightweight and not intrusive. This limited battery capacity therefore components and processors were chosen to limit energy usage.

The legal requirements for individuals to have access and control over their data collection is noted by [138]. They give the elderly individual the ability to control when they are being monitored and who has access to the recorded data. They also note that the user experience is important for the elderly person using their system. This is because designing HCI for the elderly is more difficult due to their cognitive decline, as well as the decline of their vision and physical abilities. A gateway is installed into the home of the user, which retrieves measurements from sensors. This data can be checked by the user before they give permission for it to be sent to the cloud for storage and analysis.

Data can be accessed by family or medical professionals through a web application if the data has been shared with them.

A framework for monitoring an individual's mental health is proposed by [139]. Mental health monitoring is used for monitoring sleep patterns, weight loss as part of depression and weight gain due to inactivity. They note that monitoring behaviour is more complex and dynamic than monitoring physical symptoms. Sensors for monitoring activity and behaviour are placed on the body to form a personal area network. The data from this is transmitted wirelessly to a healthcare professional. Monitoring includes vital signs, activity levels and patient symptoms. This can inform on symptoms including sleep, appetite and energy levels. Information transmitted to the healthcare professional is compared against patient records and family records to determine the authenticity of the symptoms displayed. They note that the system requires family records in order to make a diagnosis and that more works needs to be done to reduce any incorrect diagnosis.

2.5. Summary

In this section, we observed the various PA types that have been monitored in previous studies, as well as human behaviours. The sensing layer was studied, and the different types of devices that can be used for AR and for studying behaviour are observed. We looked at the processing layer and how the data collected is pre-processed through the use of Sliding Window and Bottom Up algorithms. We observed the various features that are extracted from data collected from sensors in the form of Time and Domain features, and we observed the classification techniques that are applied to the extract features. Finally, we observed the applications of PAR in natural environments in the form of Lifelogging, Assisted Living, Remote Health Monitoring and Abnormality Alerts. Many of the studies discussed highlight the important of PAR studies to improve the QoL of the aging population and those with chronic illnesses. Wearable and ambient sensors and the implementation of classification techniques in order to recognise activity aid in making long-term monitoring of individuals much more feasible.

Chapter III: Framework for Investigation of Activity Recognition in the Elderly

The main focus for investigation can be shown below in Figure 3-1. A range of activities of daily living that can be categorised as dynamic, sedentary and transitional will be selected for data collection. Activities within these categories can have varying degrees of complexity in the motions and can be difficult to classify without an appropriate time window in which the activity is performed. Ensuring that we have the appropriate set-up of sensors is essential if we are to achieve a high accuracy rate from classification. Data will be collected from the sensors and transmitted to a smart phone via Bluetooth for storage as participants perform the selected activities. Due to technological advancements made, wearable devices contain a multitude of sensors such as accelerometer, gyroscope and ECG, which have all been increasingly observed in previous works for recognising various activity types.

Raw data collected from sensor signals, contains redundant information that is filtered out in the pre-processing and processing phases. Pre-processing involves the cleaning up of the data and reducing the overall dimensions of the data set; these are then divided into appropriate time windows. Feature extraction using time domain or frequency domain techniques are applied in order to produce a better and useful representation. Classification methods will then be investigated using the extracted features in order to categorise and determine the activity types. These are then evaluated using various statistics in order to produce a measure of the classifier's overall performance.

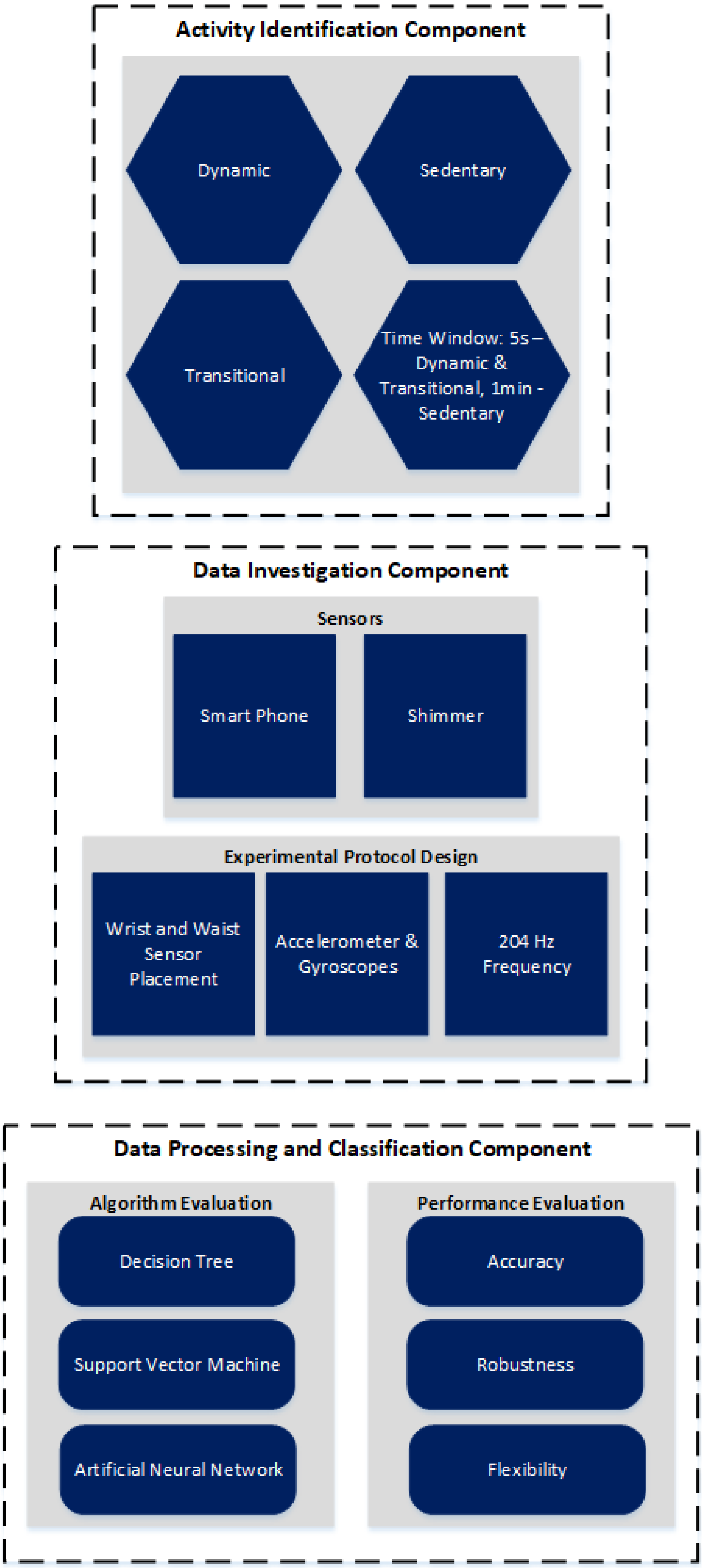

Activity Identification Component
Dynamic
Sedentary
Transitional
Time Window: 5s – Dynamic & Transitional, 1min - Sedentary
Data Investigation Component
Sensors
Smart Phone
Shimmer
Experimental Protocol Design
Wrist and Waist Sensor Placement
Accelerometer & Gyroscopes
204 Hz Frequency
Data Processing and Classification Component
Algorithm Evaluation
Decision Tree
Support Vector Machine
Artificial Neural Network
Performance Evaluation
Accuracy
Robustness
Flexibility

Figure 3-1: Proposed Investigation Framework

3.1. Background of Activity Recognition in the Elderly

As most elderly people have a desire to spend as much time in their homes as possible, it has become vital for AR to be conducted in home environments in order to maintain a sense of normality and to improve the QoL of elderly individuals [140]. Chapter 2.4 discusses the different applications of AR research and how it is integrated into individuals' lives with good results reported by surveyed studies [21], [132]. However, there are limitations, as some studies require the use of a questionnaire, presented to individuals to complete in order to provide additional resolution to activity data. Other studies require video cameras which can produce privacy concerns and are not desirable [141]. Body sensor networks are an alternative to video monitoring; however, this can produce irritability amongst the elderly, as they do not want to continuously wear sensors all over their body. [142]. Body senor networks can also not be cost-effective due to the number of sensors and the type of sensors that are utilised [20]. However, a single sensor system does not provide enough contextual data to provide a high accuracy [143]. Our aim is to produce a framework for ADL in the elderly that uses as few, low-cost sensors as possible in order to be unobstrusive and mitigate irritation for the elderly people utilzing them.

3.2. Activity Type Identification

An elderly person's ability to perform ADL with minimal assistance is an indicator into their health and wellbeing. Those that have begun to suffer with MCI, Alzheimer's, and Parkinson's or mobility issues can continue to live independently in their own homes by applying healthcare monitoring and assistive informatics [144]. Monitoring the elderly in a natural, home-based environment can provide continuous monitoring of individuals to produce larger data sets, and also provides advantages over controlled environments as it can alleviate privacy concerns that individuals may have [145]. Frequent exercise and PA are some of the most important contributors to maintaining an individual's quality of life [146]. Engaging in PA regularly by walking, jogging or performing sports activities can reduce the risk of cardiovascular diseases, obesity and diabetes [147].

The main objective of AR is to recognize human activities that are performed frequently on a daily basis in real-life environments. Type, duration and intensity of a broad range of activities are the focus of PAR assessments. Attempting to achieve a high accuracy in recognition is challenging, as human activity is complex. Individuals can perform multiple activities at once, and also have variations on how they would perform them [148]. Activities can also be ambiguous as they can belong to multiple types of activity. Multiple residents can also dwell within one environment, which adds additional challenges for smart home spaces as they need to recognize individual activities, and activities performed as a group.

It is also vital to identify the types of activity that will be performed during a study, and categorise them appropriately. Sedentary activities are a common category found in studies that involve activities such as sitting, lying down, watching TV or any activity that does not require a great deal of movement [149]. Transitional activities can indicate the ability of an individual's movement capability. As mobility becomes limited, it has been suggested that these movements become more distinct [150]. Transitions occur when an individual moves between two separate states, for example standing to sitting and sitting to lying. Transitional activities are usually ignored in studies due to the low appearance and short duration in comparison to other activities [59]. Other forms of activity can require a greater degree of locomotion or movement from the individual performing them. These have been described as dynamic activities, they involve activities such as walking, running and falling [150], [151]. These activities can produce difficulties in classification due to the variability of how they are performed between individuals.

Activities can also be visually observed through the use of a video recording. This allows researchers to match the frames of a video recording to the data set taken from the wearable sensors [148]. We can observe the sequence of poses an individual makes when performing a complex activity and compare the poses against other participant's actions. This comparative process would allow us to understand why a complex activity may produce a poor accuracy from the data collected using the wearable sensors.

Selecting the appropriate length of an activity window can have an impact on the accuracy achieved by classifiers. An activity that is performed for a short duration can achieve a high failure rate if the window is too long, however if the window is too short the activity may not be completely recorded [152], [153]. The optimal window length will differ for each activity. Walking could be performed continuously using a longer window length, while a stand-to-sit activity would only take a couple of seconds to complete, requiring a shorter window of around five seconds. The number of data points in a window and the features extracted from the set can be increased by using a higher sampling rate; however, this requires more memory, particularly for longer window lengths [154].

The aim of the framework is to record activities that fall into the dynamic, sedentary and transitional activity types to cover the different motions that a person may perform in the home environment. We will also take a video recording of each activity performed, using a separate Samsung Galaxy S6, in order to match the poses in the frames of the video to the data set collected in order to observe the individual variations of attempting an activity. Window lengths will be determined by the type of activity being performed. Longer duration activities will be recorded for longer periods such as 30 seconds or a minute, shorter activities will be recorded in a 5s time frame.

3.3. Wearable Sensor and Data Collection Investigation

Identifying the feasible devices or sensors is essential for tracking, monitoring and detecting the three activity types. Existing technologies have proved that both wearable devices and ambient sensors are widely used to monitor personal health status. Considering that our targeted group are the elderly populations and their routines in daily life, that ambient sensors or devices can involve a greater deal of work in order to deploy, we shall focus on evaluating and identifying wearable devices or sensors. Smart phones were included in this evaluation as they contain various sensors that can be used such as accelerometer and GPS but are also used to connect to wearable devices as they can connect via Bluetooth to a smart phone application.

There are a wide range of wearable PA devices that can be used in assessing PA that has become possible in research due to advances in wearable technologies [155]. Patterns in the signals from the raw acceleration data can be revealed for different activities, allowing for the use of machine learning algorithms to classify activity. Several studies attempt to classify their activity types from data collected in a laboratory setting and have structured the activity with a set time and specific instructions [156], [157].

Accelerometers have become popular for research due to their small size, low cost and ease of integration into existing sensor networks. They respond well to the frequency and intensity of human movement and can provide useful features to be used for classification of ADL or fall detection [143]. However, accelerometers alone may not be sufficient to provide the related information needed for an accurate classification. Therefore, other studies employ the use of additional sensors such as gyroscopes [158], [159], electrocardiogram (ECG) [160] and microphones [161] to improve the accuracy of their models.

Smartphones are also popular for activity recognition studies due to their low cost and the multiple sensors that can be on board the device. The advantage to using smart phone sensors to gather data, is the less obtrusive nature of it as you do not have to place multiple devices on the body [162]. However, a single smartphone device may not produce sufficient data in order to produce high accuracy of activity classification. Pairing the device with other wearable sensors using Bluetooth for data collection would improve the capabilities of the data collection network as the phone can act as a storage device until data can be transferred over to a computer or server.

Sensors can be placed on multiple positions on the body to form a body sensor network, which is practical for lab-based studies. However, in a home-based environment too many sensors would be impractical for the participants involved and ensuring the position and orientation of so many sensors would become difficult and could lead to signal variations and errors [143]. A singular wearable device, while simplifying the design of the approach, would also not provide sufficient information of the movements involved in an activity for successful activity recognition to occur [163]. Therefore using only a couple of sensors placed on the body could provide improved accuracy for the data, while keeping the number of sensors used to a minimum to avoid errors with positioning and orientation. The wrist and waist are common positions that have been used for sensor placement in several studies [81], [164], [165].

Smart phones contain multiple sensors within them; however, they are only one device with limited positioning options, usually inside a pocket of the participant. The downside is that the smartphone is not fixed into position, and therefore its position and orientation varies as an activity is performed [166]. A smart-phone could also have its position changed from the pockets to the hands between each activity relatively easily in order to improve the accuracy as a single location might not be optimally suited for collecting certain activity movements [167]. Participants may also forget to pick up a smartphone and place it into the correct position and orientation resulting in no data being collected at all [152].

The sampling rate of the sensors used can determine the accuracy achieved in AR. A high sampling rate can lead to a better accuracy, however this results in a higher energy consumption by the device due to the increased transmission rate [168]. A lower sampling rate can produce lower accuracies from classified data sets while prolonging the lifespan of the device [169]. Activities themselves can be fairly complex, as a result the optimum sampling rate can vary depending on the activity being performed. It can also vary depending on the positioning of the device used in the activity [170].

We will focus on usage of accelerometers and gyroscopes for our wearable sensors and aim to use multiple devices placed in different regions of the body for data collection. We will use two Shimmer3 IMU devices displayed in Figure 3-2, which is a low cost wearable sensor that contains both accelerometer and gyroscopes [171]. One device will be placed on the wrist and the other on the waist in order to collect sufficient data without the system being too cumbersome for the participant. The sampling rate of both sensors with be set to 204.80Hz in order to acquire enough data, particularly for complex activities, in order to improve the accuracy of implemented classifiers. A Samsung Galaxy S6 android phone will also be deployed to use for data collection, as it provides a storage solution for the data collected and is also cheap and portable. It will connect to the wearable Shimmer sensors attached to the body of a participant using Bluetooth and has an android application installed that can be used to configure the shimmer devices and control them.

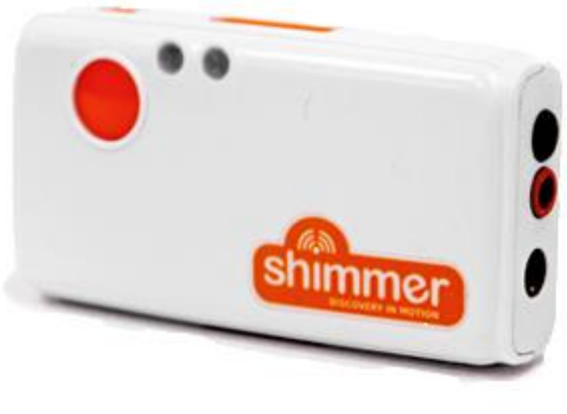


Figure 3-2: Shimmer3 IMU Device

3.4. Data Processing Investigation

In order to recognise specific human activities from a data set effectively, data has to be segmented into smaller windows to separate the various activity types occurring in a single data stream [59]. Sliding window is a popular technique for segmenting data; however, the biggest challenge with the method is due to its optimization of the window size. If the window is too small, then key information of the activity can be lost, however, if it is too large multiple activities could be added into a single window producing additional noise and resulting in a misclassification. Applying a window overlap can lead to improved accuracy during classification [63], however increasing the size of this overlap can lead to more resources being demanded.

Selecting the appropriate features to extract from a data set is important for achieving a good accuracy when applying classifiers. Feature extraction techniques help reduce the dimensionality of the data set before classification begins. Time-domain features are typically extracted from acceleration data in most studies [76], [83], [172]. Time domain features can be easily extracted from acceleration data in real-time and are simple to process, saving time, hence the popularity. However, time domain features are not able to fully express the states of motion of an activity accurately [72], [77]. Frequency domain features can also be extracted from acceleration data sets. Frequency domain features can produce differentiation between activities with simple patterns and those with complex ones leading to improved recognition [78]. Frequency domain features, however, do require the transformation from time to frequency domain which, can increase computation time of systems [173]. Choosing appropriate features is essential for achieving high accuracies when classifying data from accelerometers.

Selecting the appropriate classifier for further investigation is important as each has its own advantages and limitations. ANNs are a popular supervised learning classification method as it can be applied to various forms of challenging pattern recognition [174]. It is necessary to produce a reasonable data set for training and testing purposes of a neural network, as the algorithm requires the information in order to learn and begin to distinguish activities from each other [175]. While neural networks are powerful for function approximation and data fitting, they also suffer with the problem of overfitting where the training sample fits extremely well with the model reducing the generalization capability of the network when trying to predict [118].

Decision trees are often adopted in studies due to their low complexity and ability to interpret [176]. They are flexible in their decision-making as one can choose the different subsets of features at the internal nodes in order to choose a feature subset optimally [121]. However, there is the potential for overlap within a DT when the number of classes is large. This can increase the search time and memory space requirements. Errors can also accumulate as the tree moves throughout each level, particularly in larger trees. Designing the optimal tree can be difficult as its performance relies on the design.

SVMs are a supervised learning type of classifier, which have proven popular over recent years and have been successfully applied to many AR studies [108]. They are able to support classification as well as regression tasks and can manage multiple variables [122]. Compared to neural networks they perform better at not over generalizing larger data sets, scale well to high dimensions and can train data relatively easy. However, in order to run optimally they require a good kernel function, which adds additional complexity over their counterparts [124].

Choosing the right classifier is important, which is why we will focus on investigating the three classifiers discussed above, ANN, DT and SVM. We will choose time and frequency domain features to be extracted from the activity data we collect using wearable accelerometers as we discussed in the prior section. An appropriate overlap and segmenting technique such as sliding window will be applied during the pre-processing phase in order to match the time windows from each sensor. We will evaluate the performance of the classifiers used and compare the results from each sensor using various statistical measurements commonly used in performance evaluation of classifiers.

3.5. Summary

We have presented a framework for the investigation of activity recognition for activities of daily living. The importance of selecting appropriate activities was considered with activity duration, type and complexity being a focus. We recognise the complexity that may come with some activities due to the movements involved and potential ambiguity due to how individuals perform them. The types of sensors that can be used to capture activity data were considered, with multiple wearable accelerometers being a focus for future work. We also considered the importance of optimally positioning and orientating the sensors in order to collect data as well as the effect of a high or low sampling rate. We also discussed pre-processing steps of overlapping and sliding window, the types of features we will extract, and a brief discussion of ANN, SVM and DT classifiers. This knowledge forms our investigative framework, which will be applied to an investigation of classification techniques.

Chapter IV: Conceptual Data Collection Investigation for Activities of Daily Living

The application of wearable and ambient devices for PAR is well documented in literature with many studies seeking to provide a new methodology for PAR.

The layers displayed in Figure 4-1 are fundamental in approaching a PAR methodology. Wearable sensing devices typically contain accelerometers, gyroscopes, magnetometers, altimeters, GPS and ECG. This allows for health monitoring (weight, blood pressure, heart rate), activity tracking (walking, running, sleeping, falling), and location monitoring. Data collected from sensors is transmitted via Bluetooth or Wi-Fi to a corresponding device such as a smart phone or a laptop. The data stored on these devices is then pre-processed using a Sliding Window or Sliding Window and Bottom Up algorithm in order to break down the data into appropriate time windows.

Feature extraction methods can be applied to extract key features from data signals to produce an enhanced representation of an activity. Time domain features (Min, Max, Average, Standard Deviation) or frequency domain features (Spectral Energy, Entropy, Magnitude) using FFT can be extracted from the data. Classification methods are then applied to the extracted features for training and testing, in order to recognise and classify activities. An application layer can provide more context of the classified data for the user, such as a caregiver. This allows them to interact with the person being monitored in order for them to receive PA results or to be alerted should an abnormality occur.

This chapter presents a early concept to our approach for data collection of ADL along with data pre-processing and feature extraction techniques selected.

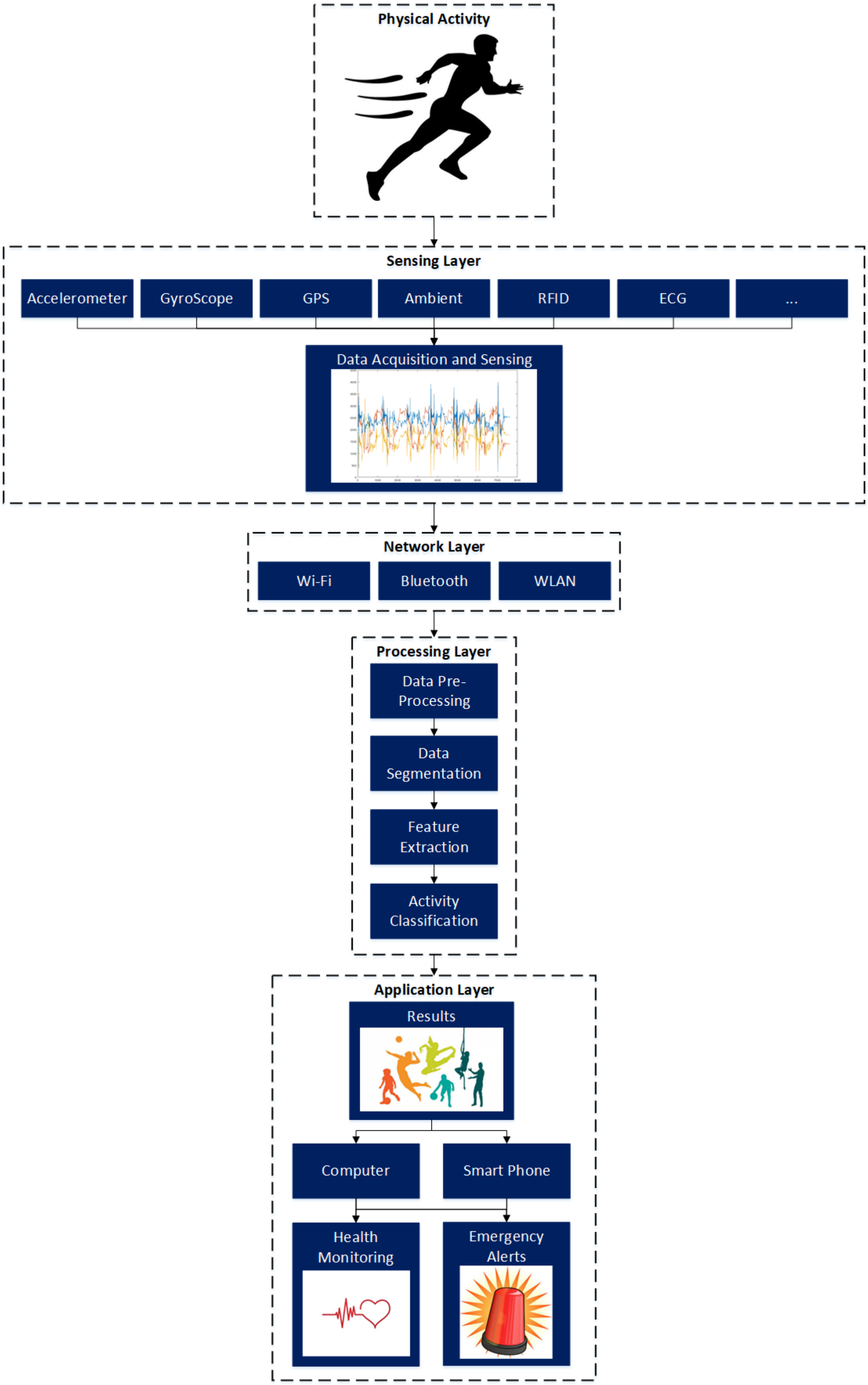


Figure 4-1: Layered procedures in PAR

4.1. Data Collection

Healthy participants under the age of 50, with no diagnosis of Mild Cognitive Impairment, or other mental illnesses or any signs of memory loss were used to collect data from fifteen activities of daily living. Data was collected using two accelerometers, which were on board two Shimmer3 IMU devices. The six participants wore a shimmer device on the wrist and on the waist as shown in Figure 4-2 below. The sampling rate of the accelerometers was set to 204.80Hz on both devices. Data was then transmitted via Bluetooth to a Samsung Galaxy S6, which was running an android application called Multi Shimmer Sync Evaluation as shown in Figure 4-3 below.

This application connects the phone to the corresponding Shimmer devices, allowing for the collection of data from multiple units. Data can be streamed and logged using the application, with data being stored on the phone. The Shimmer devices can also be configured using the application.

The machine learning process for pre-processing, feature extraction and processing of the raw sensor data can be shown in Figure 4-4.

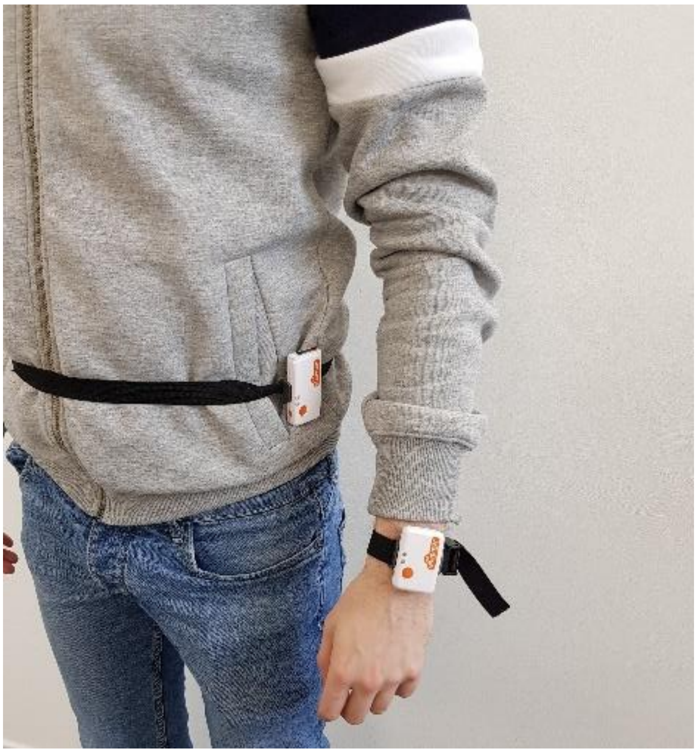
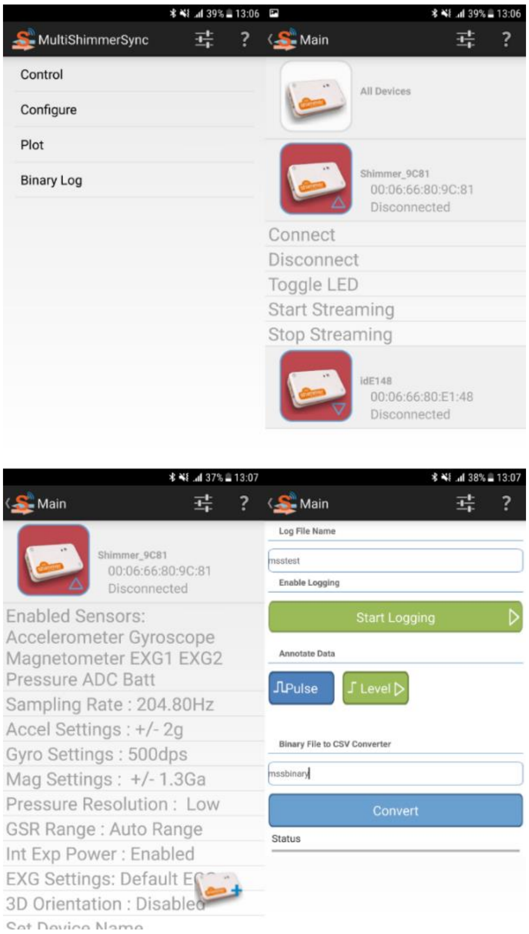


Figure 4-2: Shimmer Sensor Placement



Shimmer_9C81
00:06:66:80:9C:81
Disconnected

IdE148
00:06:66:80:E1:48
Disconnected

Enabled Sensors: Accelerometer Gyroscope Magnetometer EXG1 EXG2 Pressure ADC Batt
Sampling Rate : 204.80Hz
Accel Settings : +/- 2g
Gyro Settings : 500dps
Mag Settings : +/- 1.3Ga
Pressure Resolution : Low
GSR Range : Auto Range
Int Exp Power : Enabled
EXG Settings: Default E
3D Orientation : Disabled
Set Device Name

mssstest

mssbinary

Figure 4-3: Multi Shimmer Sync – From top left, Home Page, Control Page, Configuration Page and Binary Log

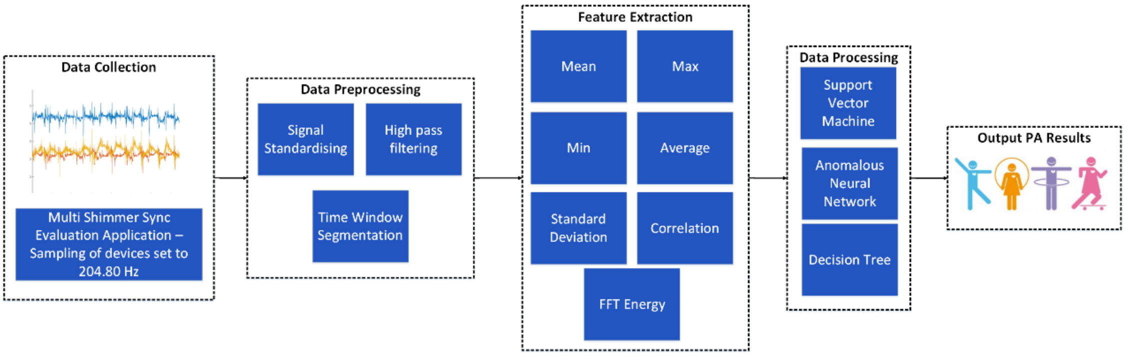


Figure 4-4: Machine learning approach to PA recognition using wearable sensors

Fifteen activities were performed by the six participants. These were split into three categories, Dynamic, Sedentary and Transitional and can be seen in Table 4-1 below. Sedentary activities were recording in one sitting, for a total of 1 minute. Transitional activities where recorded 10 times each with each activity taking 3 – 5 seconds. Walking was recorded 3 times, with a participant walking up and down the room taking 20 seconds to complete. Walking up and down stairs were recorded 5 times each taking approximately

10 seconds to complete. The remaining dynamic activities were recorded 10 times each, taking approximately 3-5 seconds to complete. These time windows were chosen in order to capture the motion of the activity, while ensuring no additional noise was captured as this could reduce classification accuracy. Raw sensor data was converted on the phone to a compatible excel format and then transferred to a desktop device for storage. An example of the raw sensor data from the wrist accelerometer is displayed in Figure 4-5. Sedentary activities display some movement in the raw data. This could be due to the participant unconsciously moving while data collection was taking place.

Table 4-1: Activities of Daily Living

Activity ID	Activity of Daily Living	Categories	Description
A1	Picking Up and placing down Object	Dynamic	An object is picked up off a surface and raised to the chest
A2	Drinking Water	Dynamic	A glass of water is raised from the chest to the lips
A3	Opening Door	Dynamic	Participant opens a door
A4	Closing Door	Dynamic	Participant closes a door
A5	Walking	Dynamic	Participant walks up and down a room
A6	Walking Upstairs	Dynamic	Participant walks up a flight of stairs
A7	Walking Downstairs	Dynamic	Participant walks down a flight of stairs
A8	Sitting	Sedentary	Participant remains in an idle sitting position
A9	Standing	Sedentary	Participant remains in an idle standing position
A10	Lying	Sedentary	Participant remains in an idle lying position for
A11	Standing to Sitting	Transitional	Participant goes from a standing to a sitting position
A12	Sitting to Lying	Transitional	Participant goes from a sitting to a lying position
A12	Lying to Sitting	Transitional	Participant goes from a lying to a sitting position
A14	Sitting to Standing	Transitional	Participant goes from a sitting to a standing position
A15	Bending Over to Standing	Transitional	Participant goes from a bent over position to a standing upright position

Figure 4-5 below displays the raw accelerometer data from the wrist accelerometer of the three walking activities performed. The three activities are all fairly similar, with each individual maintaining similar arm motions as they walk. However, walking up and down the stairs appears to be more intense with more swings of the arm being detected compared to walking normally.

The peaks on the X-axis also differ between walking and walking up and down the stairs. The height of the peaks for walking is relatively low, while the range of the peaks for up and down stairs is much larger. For climbing the stairs, this could be caused by an individual exerting more physical motion while trying to climb up.

This could also be observed in the Y-axis as the motions are not as frequent, but the peaks are more intense at times. The change in elevation for both activities is also a contributing factor for the increase in this range. One factor that could have an affect on the signal output for the wrist acceleromer is if the hand, to which the sensor is attached, is on the railing for balance, this could result in the walking up/down stairs activity to be not classified or misclassified.

Despite the differences in the intensity and frequency, the three activities are very similar to one another, this could result in potential misclassifications of the activities if the features extracted are not sufficient.

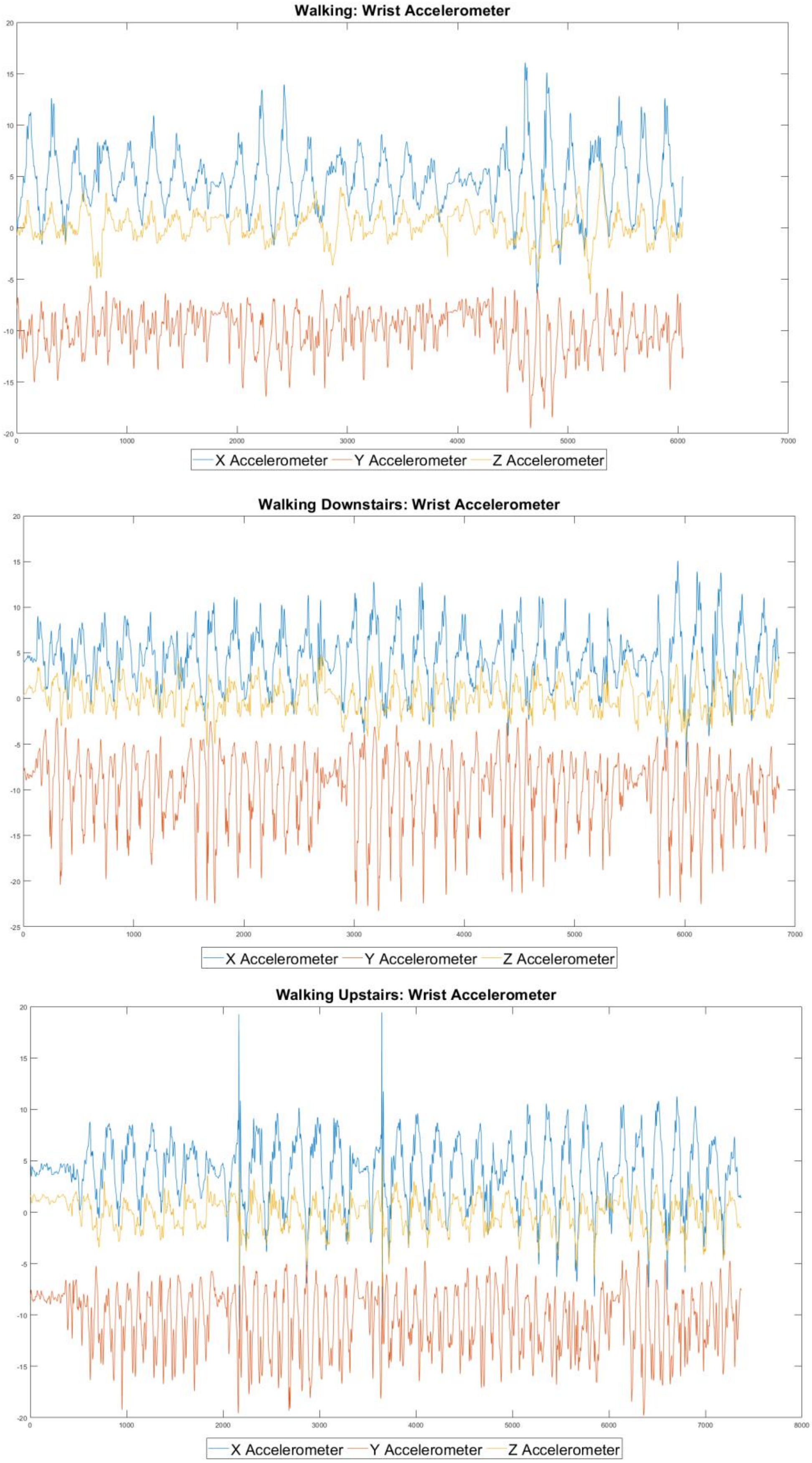
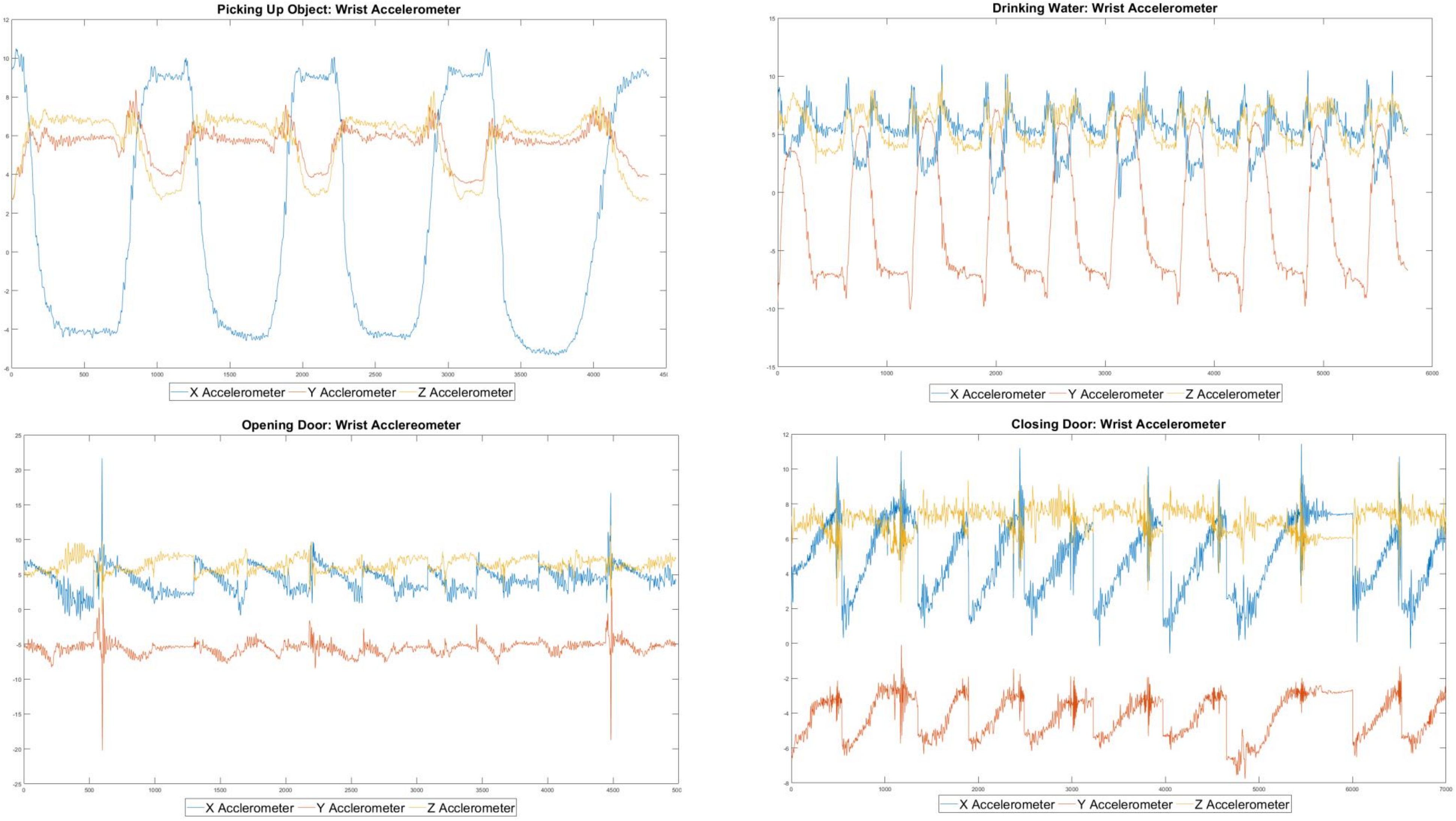


Figure 4-5: Raw sensor data from the wrist accelerometer of walking activities from the dynamic activity category.

Raw accelerometer signals from opening/closing door, picking up object and drinking water can be seen in Figure 4-6 below. The X-axis is the most defining signal produced by the accelerometer for picking up an object. This is due to the orientation of the accelerometer and the up and down motion performed when picking up and placing the object back down.

A similar pattern can be observed in the raw signal from the Y-axis, from the drinking water graph as this movement involves the participant bringing a glass towards their mouth to take a sip of water and then taking it away. The X-axis for drinking water also displays a larger range in its peaks compared to picking up an object as this activity also involves some vertical movement as the participant lifts the glass up from their chest towards their face and then back down. The Z-axis of the picking up an object raw signal displays more erratic movement compared to drinking water. This could be due to the participant being more focused on bringing a glass towards their mouth when drinking water and therefore having more control over their movement compared to picking up an object where their hand may sway slightly horizontally. The larger peaks of the Z-axis when picking up an object could be due to the participant bringing their arm back towards their side after placing the object back down in front of them.

The X-axis of the raw signals from the opening and closing door activities in Figure 4-6 are similar but reversed due to the nature of the activity. The Y-axis on closing door also shows a similar pattern to the X-axis of closing door. The X-axis is also more defined, with larger peaks compared to the X-axis of opening door. This could be because the angle of the door has been wider at the start of closing door compared to the end of opening door resulting in the participant having a greater distance to pull the door closed. The Z-axis of opening door does not have the same range of peaks compared to closing door. This is potentially due to the participant having their arm on the door handle for the start of opening door; however, the recording for closing door has captured the participant moving their arm back towards their side following the activity's completion.


Picking Up Object: Wrist Accelerometer

Drinking Water: Wrist Accelerometer

Opening Door: Wrist Acclereometer

Closing Door: Wrist Accelerometer

Figure 4-6: Raw sensor data of dynamic activities from wrist accelerometer.

Raw acceleration signals from both the wrist and the waist accelerometers for sedentary activities can be seen in Figure 4-7 below.

The wrist signal for lying down does not display many features as the arm is at rest for the activity's duration. Any peaks displayed would most likely be from the participant slightly moving their arm unconsciously. The same is true for standing and sitting; however, with standing the features displayed could be due to the participant moving slightly as they stand, causing the accelerometers to recognize movement. The features displayed from the sitting data could mean that the participant has intentionally moved their arm when it should have been at rest.

The waist signal for standing also has very few features as the participant is sedentary, however they may be unconsciously moving which is producing the small peaks in the acceleration data. Sitting has much larger peaks in comparison. Here the participant may have moved their upper body causing the sensor to move creating the higher peaks displayed. Likewise, lying has larger peaks, which display a similar repetitive motion on the X-axis. This is most likely caused by the participant breathing and the sensor being pushed slightly during the process.

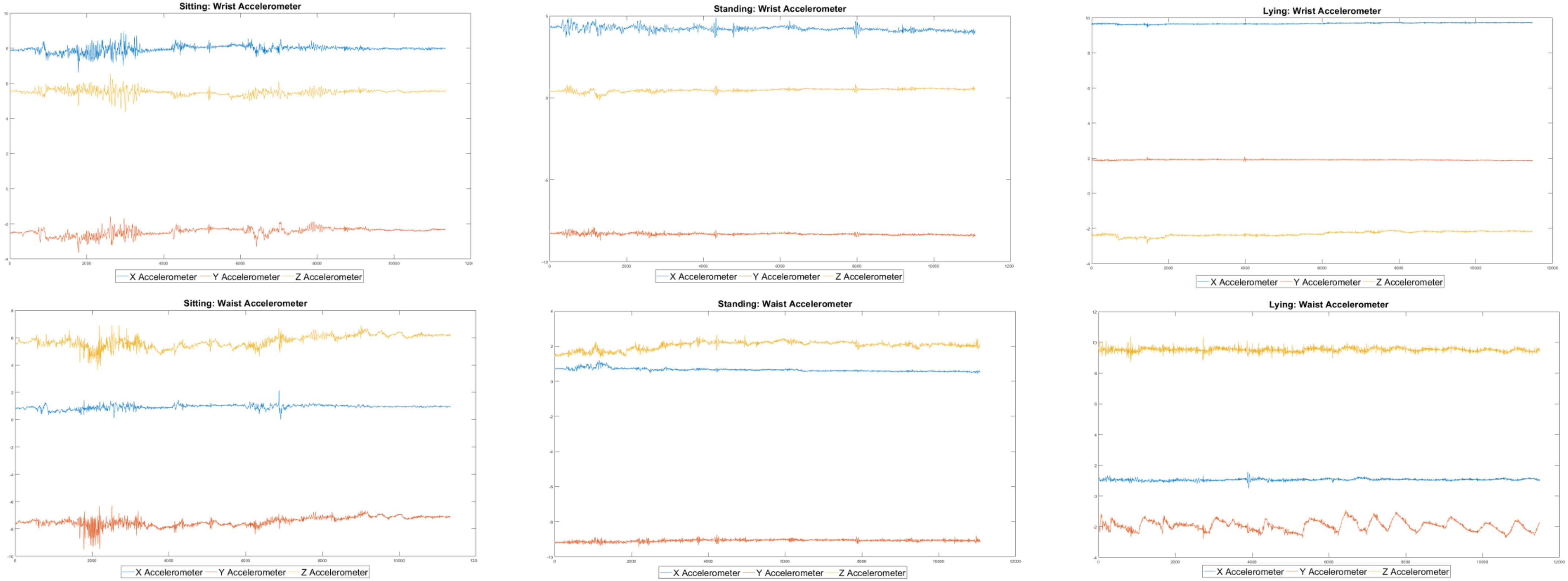


Figure 4-7: Raw sedentary data from wrist and waist accelerometer.

The raw accelerometer signals from the transitional activities are displayed in Figure 4-8 and Figure 4-9 below. These are more complex activities compared to the dynamic and sedentary activities and this can be shown by the intensity of the signals produced. The complex nature of these activities could make it difficult to classify or increase the likelihood of a misclassification if sufficient features cannot be extracted. Stand-to-sit and sit-to-stand produce a very similar X-axis, with some outliers on the peaks. The Y-axes are also similar to one another however, the Z-axes does produce different signal patterns with sit-to-stand having lower peaks than stand-to-sit. This could be due to the way the participant uses their hand to help them sit and to stand. Sit-to-Lie and lie-to-sit are not as similar to each other, the X-axis shows very different peaks and minimum and maximum ranges across the signals while the Z-axis show a bigger gap between peaks on sit-to-lie. The X-axis also shows some sudden spikes in the signal on several occasions when performing sit-to-lie; this is most likely due to the participant moving the arm slightly differently when they are using it to aid in the activity.

Bend-to-stand has a unique signal compared to the other transitional activities. The X-axis and Y-axis produce a consistent signal for each performance of the activity. However, the Z-axis does have some outliers in the signals with there being a big gap between peaks in once instance, and a much lower peak in another. This could be due to the participant moving their arm differently in order to maintain balance.

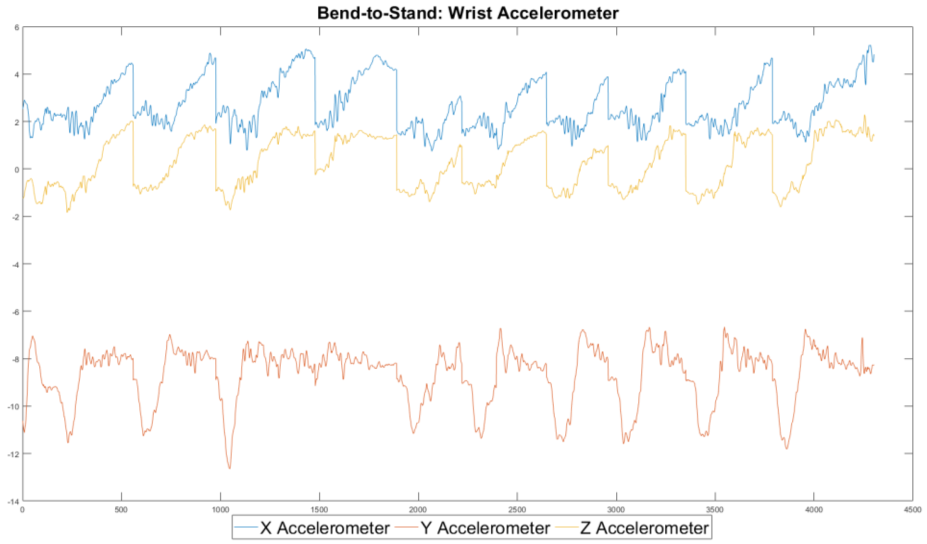


Figure 4-8: Raw transitional data for Bend-to-Stand from wrist accelerometer.

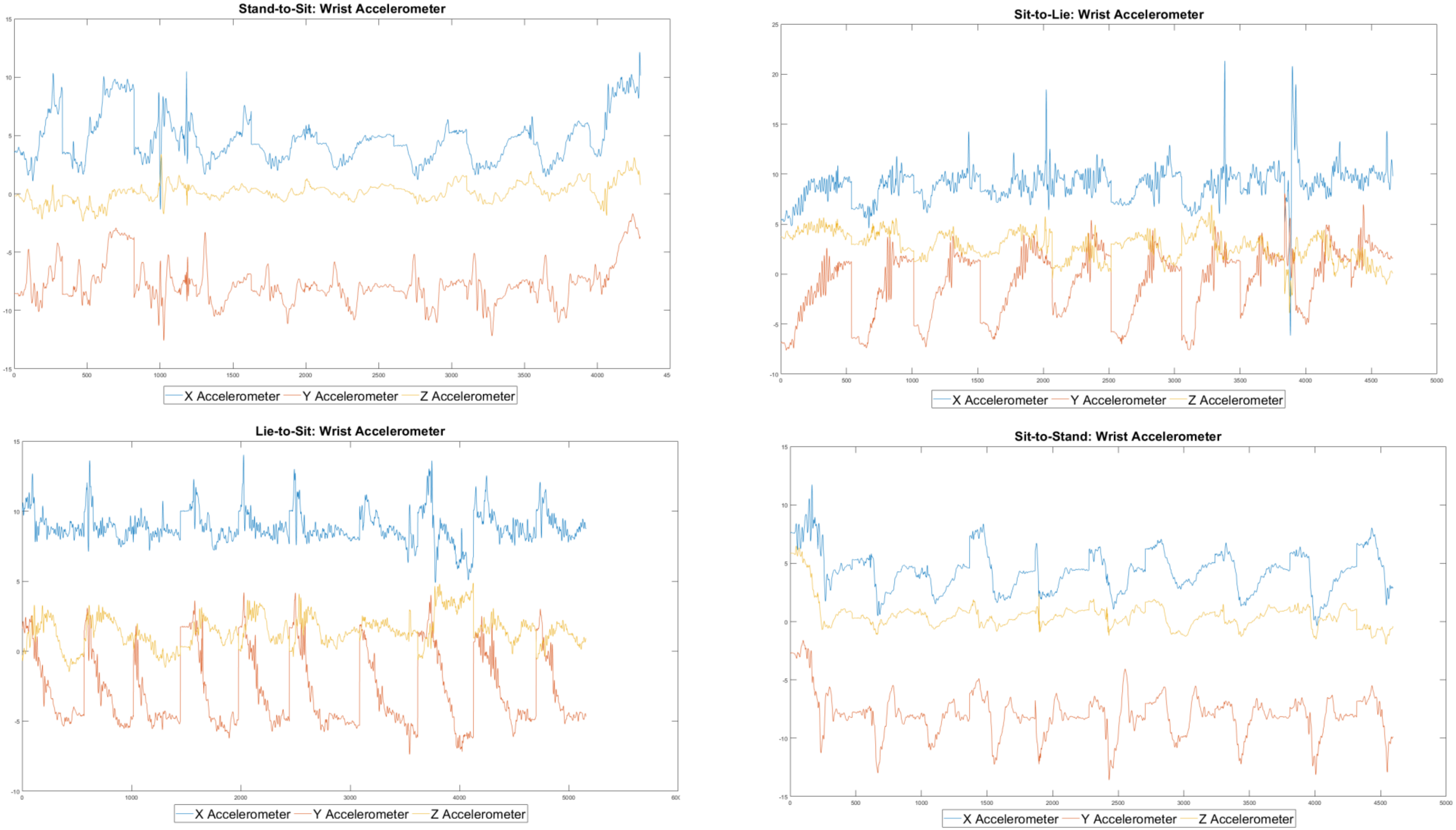


Figure 4-9: Raw transitional data from wrist accelerometer.

4.2. Data Pre-Processing

We first apply a signal magnitude vector (SMV) to the data collected from the Shimmer3 devices. SMV is used to evaluate the degree of movement and intensity from acceleration signals where x_i, y_i and z_i represent the axis of the sensor and is displayed in Equation 4-1. It is the most common value for deriving the thresholds of acceleration peaks [177].

A Savitzky-Golay filter is then applied to the data set for the purpose of smoothing the data and to increase the precision of the data [178]. An example of the filter being applied can be seen in Figure 4-10 where the filter was applied to a walking activity data set from the wrist sensor.

$$SMV = \sqrt{x_i^2 + y_i^2 + z_i^2}$$

Equation 4-1: Signal Magnitude Vector

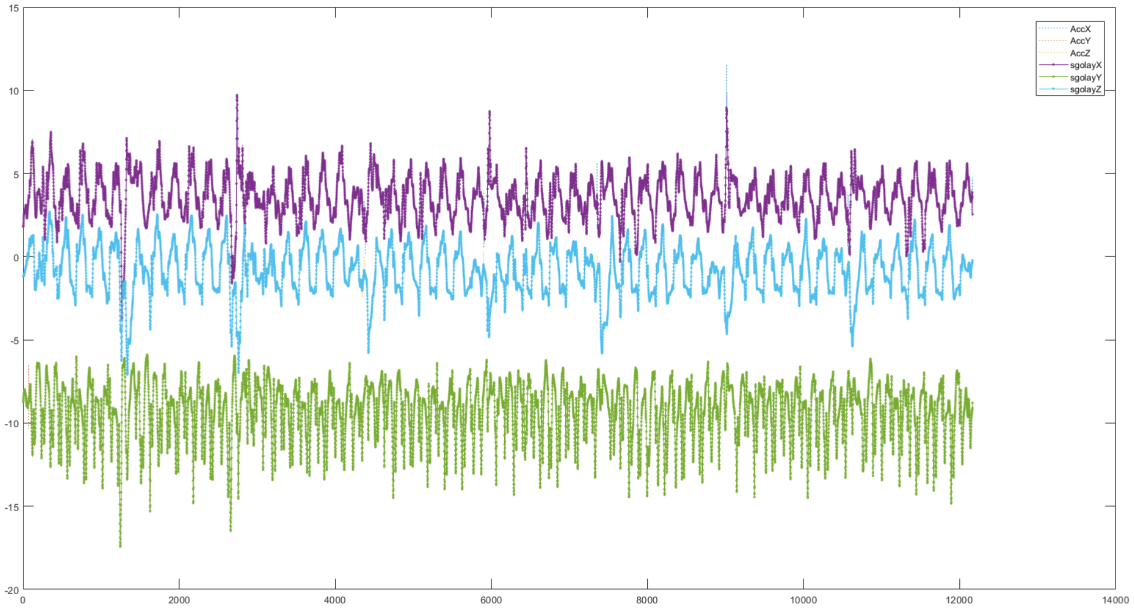


Figure 4-10: Savitzky Golay Filter applied to walking acceleration data from wrist sensor

4.3. Feature Extraction

Features are extracted from the raw accelerometer data and given a 50% overlap using sliding window. Time domain features were extracted from the raw accelerometer data collected from the two Shimmer devices as displayed in Table 4-2 below. Data sets are standardised and filtered to match the patterns of PA. The results need to be separated due to the multiple sensors consecutively running and collecting data on participant PA. Minimum, Maximum, Average, Correlation, Variance and Standard Deviation were extracted for the X, Y and Z Axis of the accelerometers. Frequency domain features were also extracted in the form of Fast Fourier Transform (FFT) Energy.

Table 4-2: Features extracted from Raw Accelerometer Data

Features Extracted	
Time Domain Features	Mean, Standard Deviation, Min, Max, Correlation, Variance, Signal Magnitude Vector
Frequency Domain	Fast Fourier Transform Energy

The feature formula for variance and standard deviation where $\bar{x}$ is the mean value of x_i:

$$\sigma^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2$$

Equation 4-2: Formula for Variance

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

Equation 4-3: Formula for Standard Deviation

The formula for the correlation coefficient where Cov is the covariance, divided by the product of σ_x (the standard deviation of x) and σ_y (the standard deviation of y).

$$\rho_{xy} = \frac{Cov(x,y)}{\sigma_x \sigma_y}$$

Equation 4-4: Formula for Correlation Coefficient

Energy is defined as the normalized sum of the absolute values of the discrete Fourier transform of the time window were F_i represents the current Fourier transformation component of x.

$$E_{(x)} = \frac{1}{N}\sum_{i=1}^{N} F_i^2$$

Equation 4-5: Formula for Energy

Table 4-3 and Table 4-4 display examples of each of the features extracted from the wrist accelerometer signals and will be used for classification of activity types. Activities that share similar values among their features may be more difficult to classify correctly as the risk of misclassification increases. This can be seen on the walking up and down stairs activities (ID 6 & 7) where many of the feature values are similar to each other such as the X maximum (maxX), Y maximum (maxY) and Z maximum (maxZ).

However, features such as XY correlation (XYcorr), XZ correlation (XZcorr) and accelerometer standard deviaton (StdACC) do highlight significantly different values, which will aid in classification. Walking (ID 5) displays very different feature values compared to walking up and down stairs. The standard deviation of X and Y are examples of these differences.

Opening and closing door (ID 3 & 4) display very similar maxY and Y minimum (minY) due to the lack of movement along that axis for the activity. MaxX, maxZ, X minimum (minX), and Z minimum (minZ) do have wider ranges between the features for the two activities as movement is concentrated on those two axes. However, the maxX, minX, accelerometer maximum (maxACC) and energy of closing door is similar to walking up and down stairs which could lead to these activities being misclassified as another of these activities. Opening and closing door also display features that are different and features that are similar to each other; this could result in classification of these two activities being difficult.

Picking up object and drinking water (ID 1 & 2) display very different features from each other despite the motions of the activities being fairly similar. This could be due to the positioning of the arm during the activity and the range of motion that is employed by the participant when performing the activity. However drinking water shares similar features with opening and closing door such as the minX, minZ, stdACC and YZ Correlation (YZcorr), increasing the likelihood of a misclassification of these activities.

The sedentary activities, sitting, standing and lying (ID 8, 9 & 10) have features that vary from each other and some that are similar due to the nature of the activities. Energy, maxACC and YZcorr, for example are very similar between sitting and lying. This could be due to the activities being isolated to specific positions where there is less chance for unconscious movements, unlike standing where the user may be moving slightly unintentionally. Standing's features do vary from the other two sedentary activities slightly, more which improves the chances of a correct classification for this activity.

For the transitional activities, stand-to-sit, sit-to-lie, lie-to-sit, sit-to-stand and bend over-to-stand (ID 11 – 15) the features do vary from each other in the tables below.

However, we can see some similarities between these activities and activities in the dynamic and sedentary types. The maxX of stand-to-sit is similar to that of closing door and walking up and down stairs. TheX standard deviation (stdX) and Y standard deviation (stdY) of sit-to-lie and lie-to-sit are relatively similar to each other along with their energy, which is also similar to sit-to-stand. These similarities will be caused by the complex nature of the transitional activities as they involve various movements in order for the participant to complete the activity. As a result, the classification of these activities will be more difficult if the classifier cannot discern a noticeable pattern for each activity.

Activity of Daily Living	maxX	min X	avg X	stdX	max Y	min Y	avg Y	stdY	max Z	min Z
Picking Up and placing down Object	1.070843	0.961209	1.013372	0.039396	0.44751	0.278591	0.359928	0.065289	0.445555	0.272174
Drinking Water	0.921251	0.536358	0.77591	0.126344	0.068851	-0.96521	-0.45592	0.330608	0.726206	0.534593
Opening Door	0.7179	0.577755	0.654284	0.037099	-0.44894	-0.60555	-0.53993	0.036193	0.555572	0.469213
Closing Door	0.462667	0.224584	0.389664	0.072204	-0.53055	-0.67913	-0.61269	0.047294	0.796428	0.59007
Walking	0.307501	-0.01551	0.154316	0.107543	-0.69114	-1.21261	-0.93073	0.158506	0.013453	-0.07549
Walking Upstairs	0.476064	0.357549	0.418743	0.031136	-0.77326	-0.88746	-0.85107	0.03667	0.175849	0.087896
Walking Downstairs	0.480517	0.389626	0.437904	0.024376	-0.77643	-0.91365	-0.84919	0.041326	0.122515	0.020885
Sitting	0.804998	0.802004	0.803564	0.000642	-0.25457	-0.25914	-0.25659	0.001068	0.570013	0.56474
Standing	0.448608	0.435844	0.442728	0.0038	-0.83568	-0.84637	-0.84296	0.002679	0.047452	0.038658
Lying	0.991743	0.980599	0.985217	0.002757	0.195598	0.18921	0.192498	0.001627	-0.24106	-0.25611
Standing to Sitting	0.407023	0.341158	0.371719	0.016788	-0.85329	-0.89341	-0.87537	0.010329	-0.00521	-0.03817
Sitting to Lying	0.649172	0.535047	0.574884	0.035841	-0.69305	-0.77438	-0.72224	0.027299	0.442463	0.360071
Lying to Sitting	1.124859	0.990486	1.0672	0.039302	0.248274	0.134211	0.184594	0.030006	0.133238	-0.07003
Sitting to Standing	0.856588	0.772284	0.787872	0.024732	-0.27221	-0.3241	-0.28472	0.01481	0.661539	0.583747
Bending Over to Standing	0.295532	0.134058	0.225231	0.06505	-0.72071	-1.13198	-0.92521	0.158025	-0.04158	-0.12653

Table 4-3: Examples of features extracted from wrist accelerometer for each activity

Activity of Daily Living	avgZ	stdZ	maxACC	minACC	avgACC	stdACC	Xycorr	Xzcorr	Yzcorr	energy
Picking Up and placing down Object	0.358446	0.071095	1.222674	1.037562	1.135717	0.076627	0.922583	0.924126	0.996877	2.750045
Drinking Water	0.633639	0.047956	1.413487	0.814922	1.137875	0.206167	-0.87932	0.100399	0.072303	3.853682
Opening Door	0.518299	0.022232	1.032294	0.944022	0.99531	0.027031	0.134241	-0.3742	0.072975	2.176248
Closing Door	0.678741	0.054095	1.099632	0.922086	0.997751	0.051354	0.453396	0.254938	0.006205	2.322664
Walking	-0.02893	0.026065	1.251634	0.692365	0.948342	0.16959	-0.7666	0.457256	0.150684	3.233593
Walking Upstairs	0.142952	0.030058	0.998576	0.864198	0.960082	0.039087	-0.16844	0.216319	-0.34629	2.130958
Walking Downstairs	0.075751	0.029359	1.016398	0.879633	0.959029	0.044994	-0.53165	0.908881	-0.39645	2.230812
Sitting	0.567145	0.001417	1.017481	1.015593	1.01647	0.000479	0.011043	-0.62916	0.511235	2.037461
Standing	0.042129	0.002433	0.957508	0.947536	0.953096	0.002583	0.213957	0.367313	0.798096	1.92327
Lying	-0.24724	0.003953	1.041965	1.028242	1.033851	0.003323	-0.53725	-0.86245	0.503353	2.092488
Standing to Sitting	-0.0204	0.010467	0.980221	0.921202	0.951385	0.014864	-0.71591	0.56412	-0.06876	1.997754
Sitting to Lying	0.39024	0.028486	1.097925	0.951831	1.00235	0.050403	-0.9532	0.941503	-0.91302	2.28985
Lying to Sitting	0.009998	0.052021	1.144863	1.007078	1.084621	0.042084	0.381488	0.473392	-0.26506	2.467668
Sitting to Standing	0.604345	0.021045	1.126755	1.012146	1.033021	0.034373	-0.86235	0.946254	-0.83029	2.304157
Bending Over to Standing	-0.06883	0.02887	1.175135	0.735567	0.955486	0.168923	-0.92582	-0.66538	0.849599	3.08268

Table 4-4: Further Examples of wrist accelerometer extracted features for each activity

4.4. Summary

This chapter presents an investigation into classification techniques that can be applied to ADLs. We considered the different types of activity that can be considered ADL and the appropriate window for recording the data.

Fifteen activities, which fall into three activity types, dynamic, sedentary and transitional, were performed by six healthy participants under the age of fifty.

We considered the different types of sensors that can be used for data collection such as accelerometer and smart phones. Data was collected using two Shimmer3 IMU devices placed on the wrist and the waist with the sampling rate set to 204.80Hz.

We discussed the raw signals of the acceleration data and highlighted discernible features that can be observed. Data pre-processing was performed using the Savitzky-Golay filter and by obtaining the signal magnitude vector. Time domain and frequency domain features were extracted from the raw accelerometer data and given a 50% overlap.

Chapter V: Machine Learning Techniques for Activity Classification

Following the data collection and feature extraction in Chapter 4, we implemented three classifiers ANN, DT and SVM to classify the extracted features. These classifiers produced an overall accuracy and several statistics that could be used to evaluate the performance of each classifier and its ability to classify individual activities. This performance indicated which classifier we should focus on, in order to improve the accuracy further in the Case Study. This results in the introduction of our Convolutional Neural Network which is discussed in detail in this chapter.

5.1. Activity Classification

Using the extracted features produced from the raw accelerometer data collected from wrist and waist devices, supervised learning techniques were applied in order to classify 15 activities, split into three activity types: dynamic, sedentary and transitional. This was performed as an investigation into machine learning techniques in order to compare the performance of the chosen classifiers, DT, SVM and ANN. The sampling rate of the two accelerometers was set to 204.80Hz. Data sets were partitioned, with 10% of data being used for testing and 90% for training. A confusion matrix was produced for each classifier applied to each data set from the two devices; an example is display in Figure 5-1. From the confusion matrix, the following values can be obtained:

- True Positive (TP) – The number of instances that were correctly given a positive classification
- True Negative (TN) – The number of instances that were correctly given a negative classification.
- False Positive (FP) – The number of instances that were incorrectly given a positive classification.
- False Negative (FN) – The number of instances that were incorrectly given a negative classification.

Using these values, we can calculate the following statistics:

- Accuracy – The most common type of statistic used in activity classification. It can be used to provide a summary of the overall performance of the classifiers. It can be defined as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Equation 5-1: Formula for calculating Accuracy

- Precision – This is a description of random errors and produces a measure of statistical variability. It is defined as:

$$Precision = \frac{TP}{TP + FP}$$

Equation 5-2: Formula for calculating Precision

- Recall – Also known as sensitivity or the true positive rate, this calculates the proportion of actual positives correctly identified. It can be defined as:

$$Recall = \frac{TP}{TP + FN}$$

Equation 5-3: Formula for calculating Recall

- Specificity – Also known as the true negative rate. This calculates the proportion of actual negatives that were correctly identified. It can be defined as:

$$Specificity = \frac{TN}{TN + FP}$$

Equation 5-4: Formula for calculating Specificity

- F-Score – This considers both precision and recall in order to compute the test's accuracy. It is defined as:

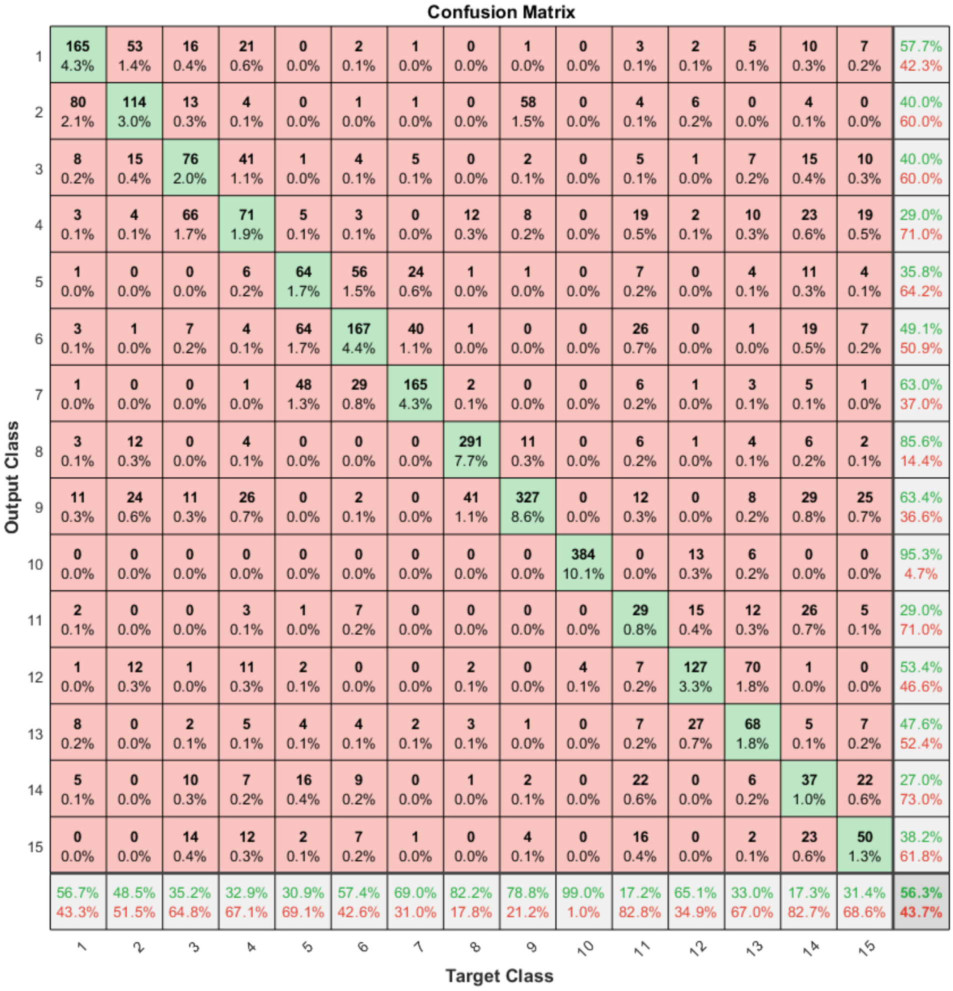
$$2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

Equation 5-5: Formula for calculating the F-Score

- Matthew Correlation Coefficient (MCC) – This takes into account both true and false positives and negatives and is a measure of the quality of binary classifications. It is used to measure the performance of a classification model It is defined as:

$$MCC = \frac{TP \times TN - FP\ \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Equation 5-6: Formula for calculating MCC



Confusion Matrix

Output Class

Target Class

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1	165 4.3%	53 1.4%	16 0.4%	21 0.6%	0 0.0%	2 0.1%	1 0.0%	0 0.0%	1 0.0%	0 0.0%	3 0.1%	2 0.1%	5 0.1%	10 0.3%	7 0.2%	57.7% 42.3%
2	80 2.1%	114 3.0%	13 0.3%	4 0.1%	0 0.0%	1 0.0%	1 0.0%	0 0.0%	58 1.5%	0 0.0%	4 0.1%	6 0.2%	0 0.0%	4 0.1%	0 0.0%	40.0% 60.0%
3	8 0.2%	15 0.4%	76 2.0%	41 1.1%	1 0.0%	4 0.1%	5 0.1%	0 0.0%	2 0.1%	0 0.0%	5 0.1%	1 0.0%	7 0.2%	15 0.4%	10 0.3%	40.0% 60.0%
4	3 0.1%	4 0.1%	66 1.7%	71 1.9%	5 0.1%	3 0.1%	0 0.0%	12 0.3%	8 0.2%	0 0.0%	19 0.5%	2 0.1%	10 0.3%	23 0.6%	19 0.5%	29.0% 71.0%
5	1 0.0%	0 0.0%	0 0.0%	6 0.2%	64 1.7%	56 1.5%	24 0.6%	1 0.0%	1 0.0%	0 0.0%	7 0.2%	0 0.0%	4 0.1%	11 0.3%	4 0.1%	35.8% 64.2%
6	3 0.1%	1 0.0%	7 0.2%	4 0.1%	64 1.7%	167 4.4%	40 1.1%	1 0.0%	0 0.0%	0 0.0%	26 0.7%	0 0.0%	1 0.0%	19 0.5%	7 0.2%	49.1% 50.9%
7	1 0.0%	0 0.0%	0 0.0%	1 0.0%	48 1.3%	29 0.8%	165 4.3%	2 0.1%	0 0.0%	0 0.0%	6 0.2%	1 0.0%	3 0.1%	5 0.1%	1 0.0%	63.0% 37.0%
8	3 0.1%	12 0.3%	0 0.0%	4 0.1%	0 0.0%	0 0.0%	0 0.0%	291 7.7%	11 0.3%	0 0.0%	6 0.2%	1 0.0%	4 0.1%	6 0.2%	2 0.1%	85.6% 14.4%
9	11 0.3%	24 0.6%	11 0.3%	26 0.7%	0 0.0%	2 0.1%	0 0.0%	41 1.1%	327 8.6%	0 0.0%	12 0.3%	0 0.0%	8 0.2%	29 0.8%	25 0.7%	63.4% 36.6%
10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	384 10.1%	0 0.0%	13 0.3%	6 0.2%	0 0.0%	0 0.0%	95.3% 4.7%
11	2 0.1%	0 0.0%	0 0.0%	3 0.1%	1 0.0%	7 0.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	29 0.8%	15 0.4%	12 0.3%	26 0.7%	5 0.1%	29.0% 71.0%
12	1 0.0%	12 0.3%	1 0.0%	11 0.3%	2 0.1%	0 0.0%	0 0.0%	2 0.1%	0 0.0%	4 0.1%	7 0.2%	127 3.3%	70 1.8%	1 0.0%	0 0.0%	53.4% 46.6%
13	8 0.2%	0 0.0%	2 0.1%	5 0.1%	4 0.1%	4 0.1%	2 0.1%	3 0.1%	1 0.0%	0 0.0%	7 0.2%	27 0.7%	68 1.8%	5 0.1%	7 0.2%	47.6% 52.4%
14	5 0.1%	0 0.0%	10 0.3%	7 0.2%	16 0.4%	9 0.2%	0 0.0%	1 0.0%	2 0.1%	0 0.0%	22 0.6%	0 0.0%	6 0.2%	37 1.0%	22 0.6%	27.0% 73.0%
15	0 0.0%	0 0.0%	14 0.4%	12 0.3%	2 0.1%	7 0.2%	1 0.0%	0 0.0%	4 0.1%	0 0.0%	16 0.4%	0 0.0%	2 0.1%	23 0.6%	50 1.3%	38.2% 61.8%
	56.7% 43.3%	48.5% 51.5%	35.2% 64.8%	32.9% 67.1%	30.9% 69.1%	57.4% 42.6%	69.0% 31.0%	82.2% 17.8%	78.8% 21.2%	99.0% 1.0%	17.2% 82.8%	65.1% 34.9%	33.0% 67.0%	17.3% 82.7%	31.4% 68.6%	56.3% 43.7%

Figure 5-1: Confusion matrix of ANN classifier applied to Waist Accelerometer

The results from the classifiers can be shown in Figure 5-2, Figure 5-3 and Figure 5-4. ANN produced an accuracy of 56.3% on the waist accelerometer data and 29.1% on the wrist. DT produced an accuracy of 46.9% on the waist and 49.9% on the wrist. SVM produced an accuracy of 37.3% on the waist, and 31.4% on the wrist. The highest accuracy for classification from all three classifiers was for the sedentary activities. However, SVM struggled to classify sitting, a relatively straightforward activity, while ANN and DT where able to achieve a good recall. SVM most likely misclassified sitting as another sedentary activity. This is most likely to due to the simplicity of the activity compared to dynamic and transitional activities. However, we can see that the results from ANN applied to the waist data shows sitting and standing to have a lower accuracy compared to lying. As the activities are similar to each other, they could have been incorrectly classified as lying. Classifiers struggled with dynamic and transitional activities from the wrist accelerometer. However, DT did manage to have some success with several dynamic activities. Activities such as opening door and closing door are very similar to each other, and complex. This could be

]]>

the reason why the sensors have struggled to identify positive recalls and why they have had more success with specificity.

Walking activities also have a similar problem with the wrist accelerometer, as arm movements while walking, will remain similar, even when walking up or down a staircase. Participants will also perform transitional activities with some degree of variation, which makes these activities more difficult to classify. Specificity, again, was the most successful statistic for SVM and DT result sets. However, ANN did have some success with recall on transitional and dynamic activities from the waist results. The waist accelerometer also has greater levels of precision compared to the other classifiers and to the wrist sensors. It also has better performance compared to the other classifiers as indicated by the MCC score. DT also has a better MCC score on sedentary activities. However, performance on transitional and dynamic activities is pretty low, particularly on SVM. One reason for this could be due to the small sample size of data that we had available in order to train the algorithms. Further data acquisition could help in training the more complex activities that are in the data set.

Figure 5-2 below displays the statistics that were generated from the ANN classifier for the wrist and waist sensors. We can observe the lying activity on the waist sensor achieves a high result for each statistic. This indicates that the true positives were correctly identified as positive, giving a high recall and indicates a good probability of lying being classified correctly with the high precision. This results in a high F-Score, producing a good accuracy for this activity. Sedentary activities display a higher statistic over transitional and dynamic activities with transitional activities struggling to achieve a good recall on true positive values. The wrist statistics display a low recall and precision for dynamic and transitional activities. Specificity was average, indicating that the negatives were, on average, correctly classified as negative. However, the walking up and down stairs activities did achieve a higher recall and precision which show that the ANN had some success in correctly classifying these activities. Stand-to-sit had no recall or precision; however, it did achieve a high specificity, showing the model could classify negatives correctly.

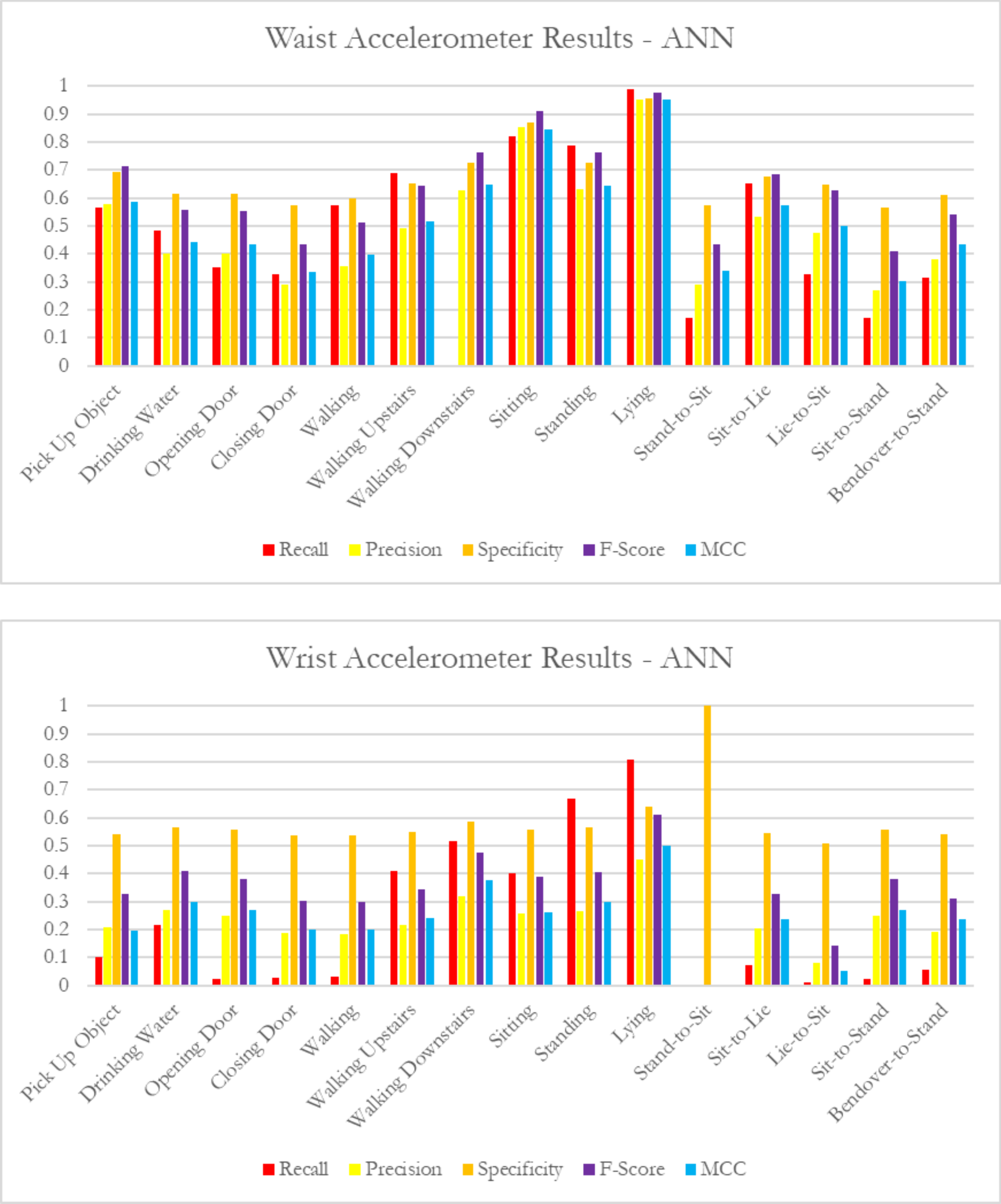
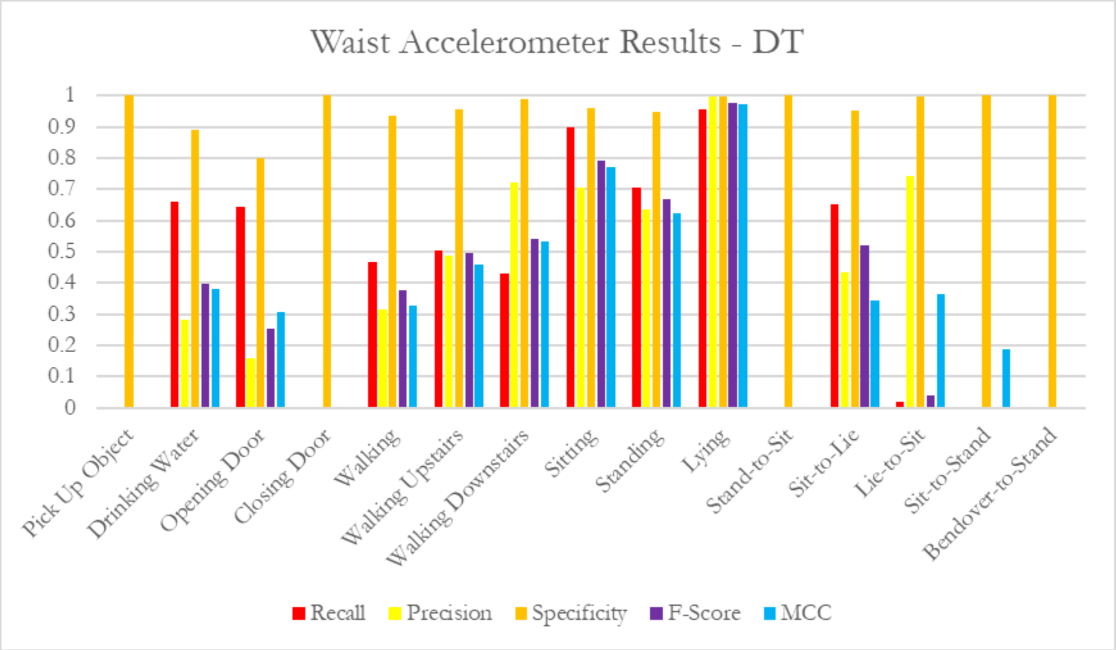


Figure 5-2: Results from ANN classifier

Figure 5-3 below displays the statistics that were generated from the DT classifier for the wrist and waist sensors. Several and transitional activities display no recall or precision for both the wrist and waist sensors. Sedentary activities achieved a high result for each statistic, indicating the model had a high probability of classifying these activities successfully. Remaining dynamic and transitional activities that displayed recall and precision, only achieved low to average results. Compared to ANN the ability for the DT to classify was inconsistent, as the DT could classify better on the wrist for some activities but for others it could not classify the activity at all.



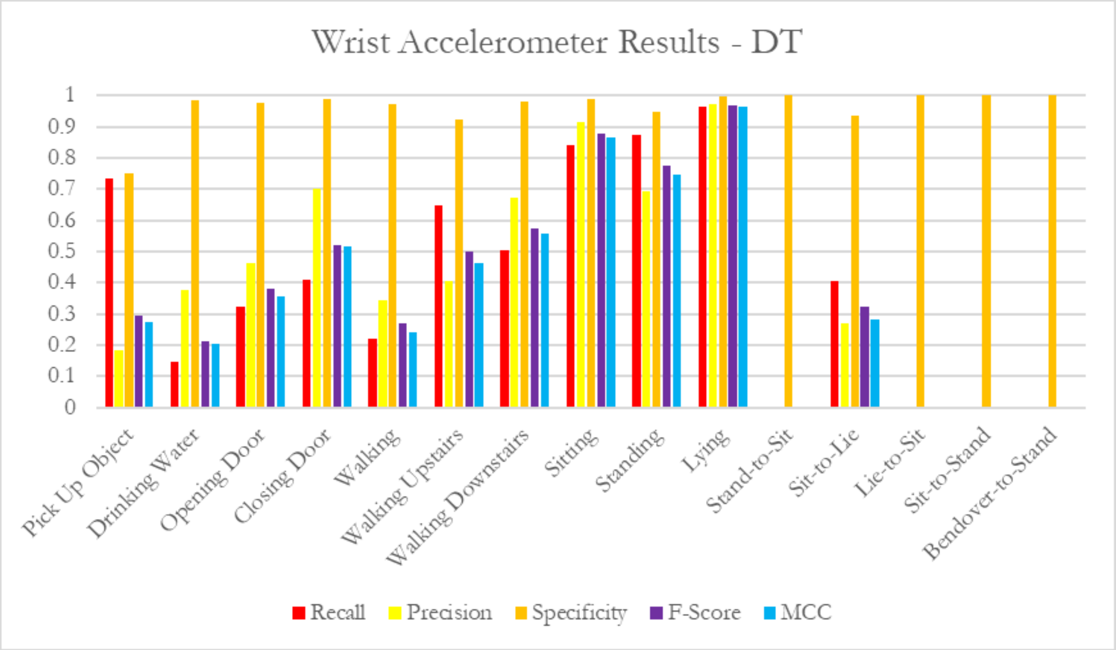
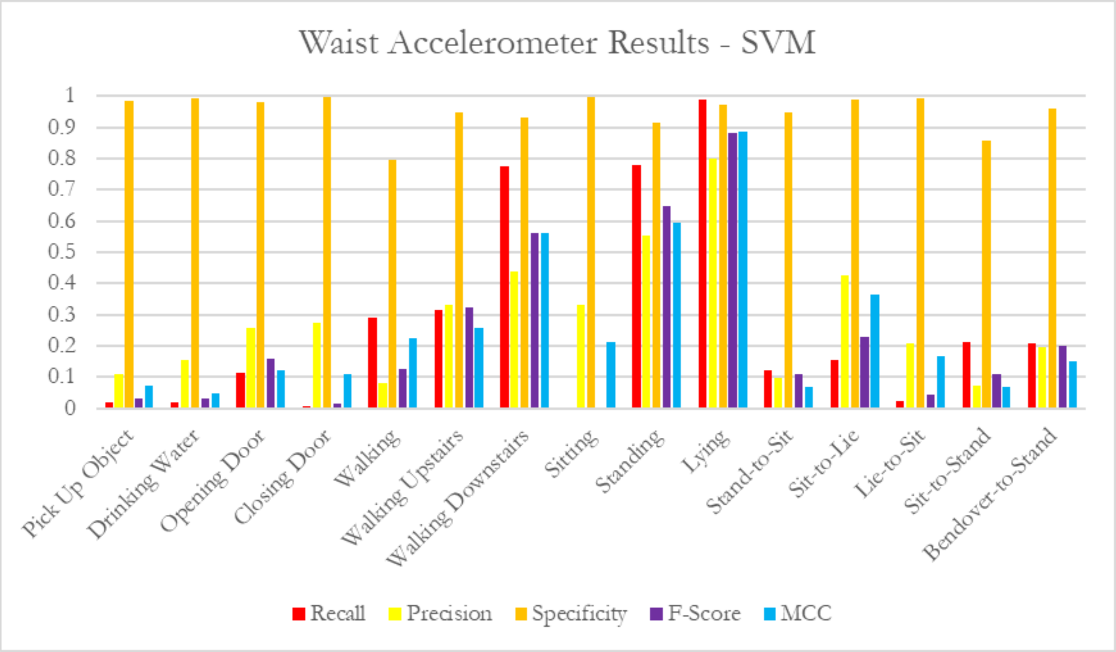


Figure 5-3: Results from DT classifier

Figure 5-4 below displays the statistics that were generated from the SVM classifier for the wrist and waist sensors. Both sensors show a low recall on all activities except lying, standing, and walking downstairs. This indicates that the model could not be trained to predict the true positive values however, it does display a high specificity indicating it could identify the true negative values. Compared to ANN and DT the performance from SVM is significantly poorer and may not be the best approach going forward.



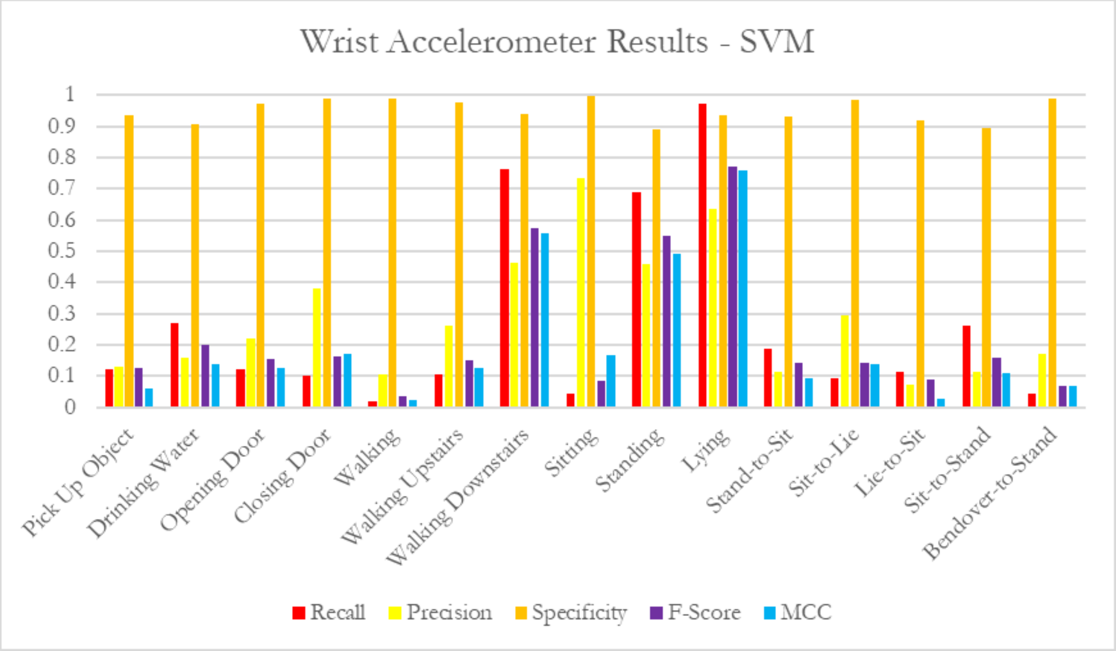


Figure 5-4: Results from SVM classifier

5.2. A Convolutional Neural Network approach to activity classification

Following the initial study using three machine learning techniques to classify physical activities discussed above, we decided to produce a model using a convolutional neural network to classify activity data, which we collected from elderly people with memory issues or a diagnosis of Mild Cognitive Impairment (MCI). We chose this method as the neural network displayed the highest accuracy for classification on the activity data collected in our previous data set. Therefore, we decided to improve upon that accuracy by refining our data collection process and implementing a model using the latest tools and libraries. In this section we will discuss the model itself and its structure. For the model we chose to use a 1-Dimensional Convolutional Neural Network (1-D CNN). 1-D CNN are good for analysis time series of sensor data original data over a fixed period. They can extract the features from these signals to create a feature map and allows the layer to make strong assumptions about the relationships. They have proven to be successful in image recognition and classification, therefore, we felt this was the best approach to move forward with following the results from the initial ANN that was used.

The subsequent chapters (Chapters 6 and 7) will contain details of the case study, the discussion of the results and the evaluation of the CNN model that we discuss here.

We developed a neural network in Python, taking advantage of Tensorflow [179] and Keras [180] libraries in order to produce our model for classification as displayed in Figure 5-5 below. As our aim is for the model to be suitable for a clinical setting, we believe that a Deep Learning approach to this model is not required, and therefore running the model on the GPU is not necessary. Achieving a high accuracy with few layers, and in a reasonable time could allow clinicians to continuously deploy the model without the need for the hardware behind it.

We applied a 1D Convolution layer for the first two layers of our network, which creates a convolution kernel that is convolved with the input to the layer over a single spatial dimension to produce the layer outputs and is suitable for activity recognition on data collected from accelerometers. This is represented in Figure 5-5 by the input and output values, the initial input shows a vector containing the 68 features from a single data point. The new shape of the data can then be seen in the output matrix after the neuron has been activated. After each convolution layer, we apply a pooling layer in order to down sample the input. Pooling is based on the "sliding window" concept and will apply a statistical function over the values within the designated window, the convolution kernel. We apply the Max pooling function for our pooling method in order to extract the finer features and perform a size reduction on the data. We also apply a drop out layer after the second convolution, where neurons are randomly selected to be ignored during the training process. This results in their contribution towards the training, being temporarily removed on the forward pass, meaning the network is less sensitive to the weights of neurons and will perform better at generalization. The data is then flattened during the flatten layer which produces a shape equal to the number of elements contained within the input in a single dimension. The final two layers of the network are dense layers which will take the features and perform a classification prediction, producing the output for the model.

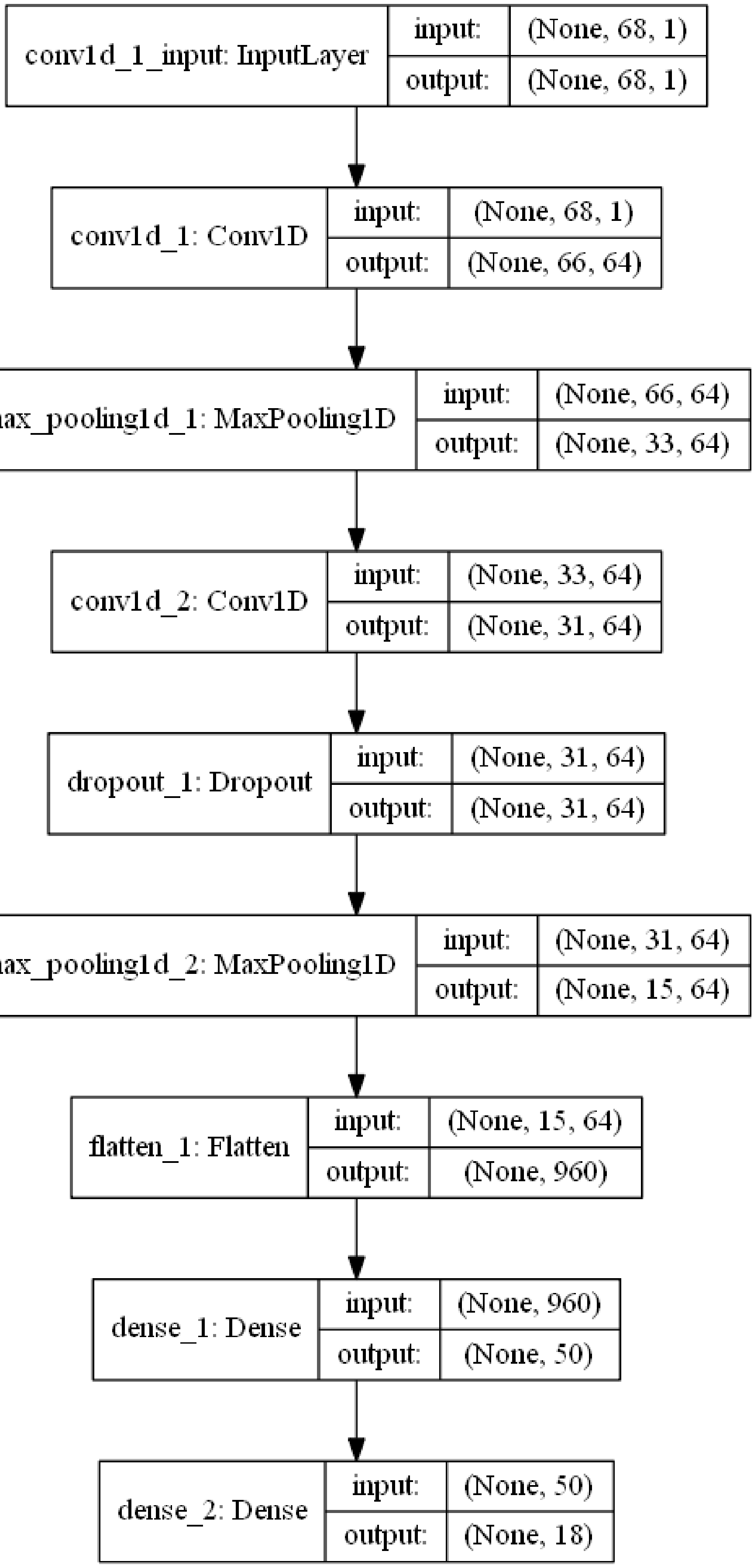

conv1d_1_input: InputLayer | input: (None, 68, 1) | output: (None, 68, 1)
conv1d_1: Conv1D | input: (None, 68, 1) | output: (None, 66, 64)
max_pooling1d_1: MaxPooling1D | input: (None, 66, 64) | output: (None, 33, 64)
conv1d_2: Conv1D | input: (None, 33, 64) | output: (None, 31, 64)
dropout_1: Dropout | input: (None, 31, 64) | output: (None, 31, 64)
max_pooling1d_2: MaxPooling1D | input: (None, 31, 64) | output: (None, 15, 64)
flatten_1: Flatten | input: (None, 15, 64) | output: (None, 960)
dense_1: Dense | input: (None, 960) | output: (None, 50)
dense_2: Dense | input: (None, 50) | output: (None, 18)

Figure 5-5: Neural Network model using 1D Convolutional layers and Dense layers.

Three activation functions, ReLu, Sigmoid and Tanh, were chosen to be implemented on the Convolution layers and first Dense layer of the model. Softmax was implemented on the final Dense layer of the model. Each activation function will be paired up with Softmax on the final layer to produce an evaluation of the model. The metrics produced will highlight the performance of the model which will be compared against other iterations of the model which will be using another of the activation functions mentioned above paired with Softmax.

The rectified linear unit (ReLu), is one of the most commonly used activation functions for neural networks. It has a linear identity for all positive values and will zero any negative values. This means that it is cheap to compute as there is no complicated maths, therefore meaning a faster computation time. It also doesn't become saturated when x becomes too large, and doesn't suffer with the issue of a vanishing gradient [181]. This allows for ReLu to converge at a faster rate. ReLu and the derivative of ReLu can be defined as:

$$f(x) = \max{(0, x)}$$
$$f'(x) = \begin{cases} 0 \; if \; x < 0 \\ 1 \; if \; x > 0 \end{cases}$$

Equation 5-7: ReLu activiation function and its derivative

The Sigmoid activation function is non-linear in nature which has a range of 0 and 1 and produces an S shaped curve for a smoother gradient. The bounding range of the function should prevent the activation from exploding. It's a differentiable function, allowing us to find the slope of the curve at any two points. However, it generally has a slow convergence rate and can suffer from a vanishing gradient. The input values are saturated which can have the potential to kill the gradient [182]. Values that are strongly negative can cause the sigmoid activation to become stuck, this can be prevented by using a stable version of the function. Sigmoid and the derivative of sigmoid can be defined as the following:

$$\sigma(x) = 1/\big(1 + \exp{(-x)}\big) = \exp(x)/\,(\exp(x) + 1)$$
$$\sigma'(x) = \ \sigma(x) \ \cdot \ \big(1 - \ (x)\big)$$

Equation 5-8: Numerically stable Sigmoid activation function and its derivative

The Hyperbolic Tangent Function (Tanh) has its output centred at zero as its range is -1 to 1 and can be interpreted as probabilities. It is capable of producing a strong gradient in an S shape similar to sigmoid. Negative inputs will map to negative outputs in tanh and the zero-valued inputs will be mapped to near zero values, reducing the chances of the network becoming stuck during training. Tanh also suffers with the issue of vanishing gradients during activation [183]. Tanh and the derivative of Tanh can be explained as the following:

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$tanh'(x) = 1 - tanh(x)^2$$

Equation 5-9: Tanh activation function and its derivative

The Softmax activation function takes the exponent of each output and normalizes each number by the sum of the exponents in order to have the output vector totalling to 1. The output is a distribution of probability, telling us the probability of classes that are true. It is commonly used on the last layer of a neural network such as a CNN [184]. Softmax and Softmax Derivative can be defined as the following:

$$s(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

$$d_j s_i = s_i\left(\delta_{ij} - s_j\right)$$

Equation 5-10: Softmax activation function and its derivative

Loss was calculated using the sparse categorical cross entropy function, which allows for the classification of multiple classes [185]. This applies to our data set, as an example in the data set can only belong to one class. The distributions of predictions are compared against the distribution of the true values with the results being a value of one of the classes (0 – 17). A false classification will be if an example is associated with the wrong class, while a true classification will be if the example is classified correctly. This approach can save memory as well as computation time due to the use of a single integer rather than a vector when predicting a class. Sparse categorical cross entropy can be defined as the following where $\hat{y}$ is the predicted value:

$$L(y, \hat{y}) = -\sum_{j=0}^{M}\sum_{i=0}^{N}\left(y_{ij} * log\left(\hat{y}_{ij}\right)\right)$$

Equation 5-11: Forumla for Sparse Categorical Cross Entropy

For our optimizer we chose Adam, an adaptive learning rate optimization algorithm that's been designed for training neural networks. It is adaptive in the sense that it computes the individual learning rate for different parameters from the estimated first and second moments of the gradients. It could be viewed as a combination of Root Mean Square Prop (RMSprop), as it uses squared gradients to scale the learning rate, and Stochastic Gradient Descent with momentum as the average of the gradient is moved instead of the gradient itself [186]. Adam works well with sparse gradients, it does not require a stationary objective and it is computationally efficient while having few memory requirements.

The moving averages are estimates of the first gradient moment, the mean, and the second moment, the uncentred variance. These moving averages are initialized as vectors of 0's, leading to moment estimates that are biased towards zero when the decay rate is small or during the initial timestep. The moving averages and the gradient and squared gradient are defined below, where m is the mean moving average, v is the uncentred variance average and g is the current mini-batch gradient.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

Equation 5-12: Adam fromula for moving averages

In order to deal with the bias towards zero, Adam deploys bias correction and is then employed on the momentum estimators. The first row of the moving average formula expands the moment, in this example we use m. At this point we wish to know the expected value of the exponential moving average at timestamp t and how it relates to the true first moment $E[g_i]$ in order to correct the discrepancy between the two. If the moment is stationary then ζ remains at 0, otherwise it is kept small as the exponential moving average assigns small weights to gradients too far in the past. The bias correction for the first momentum estimator can be defined as the following:

$$E[m_t] = E\left[(1 - \beta_1)\sum_{i=1}^{t} \beta_1^{t-i} g_i\right]$$
$$= E[g_i](1 - \beta_1)\sum_{i=1}^{t} \beta_1^{t-i} + \zeta$$
$$= E[g_i](1 - \beta_1) + \zeta$$

Equation 5-13: Adam formula for bias correction

The resulting term from the previous formula is then used to divide the first and second moments in order to correct the bias. The corrected estimators for the two moments can be defined as the following:

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1^t}$$
$$\widehat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

Equation 5-14: Adam definition of corrected estimators

In order to scale the learning rate for each individual parameter, the weights are updated. This can be done using the following formula where w is the weights and η is the step size.

$$w_t = w_{t-1} - \eta \frac{\widehat{m}_t}{\sqrt{\hat{v}_t + \epsilon}}$$

Equation 5-15: Adam formula for weight updates

5.3. Summary

Three classifiers, ANN, DT and SVM were implemented and applied to the extracted features to recognise the collected activities. ANN provided the best accuracy from its waist accelerometer with 56.3% but struggled to classify activities from data collect using the wrist accelerometer, achieving only 29.1%.

ANN and DT were fairly successful with sedentary activity types as they are more straightforward, however, SVM struggled to classify sitting, most likely misclassifying it as lying or standing. Classifiers struggled with dynamic and transitional activities from the wrist accelerometer with DT having some success with several dynamic activities. This could be due to the activities being complex or perhaps having similar characteristics to other activities and therefore being misclassified.

We display our convolutional neural network model that is comprised of convolutional and dense layers and uses categorical cross entropy for loss.

Three activation functions Relu, Sigmoid and Tanh are displayed for use in evaluation of the model with Softmax being used in the final layer for classification. The Adam optimizer was chosen to be used for weight adjustment.

Chapter VI: Case Study

By establishing the structure of our framework and running a trial as a proof of concept by collecting data from participants under the age of 50, we can then move forward and perform a case study and recruit participants from our targeted elderly demographic. This will follow the same structure as shown in Figure 4-1, however, data collection has been improved by incorporating the gyroscope on the Shimmer device sand extracting additional features from the raw data. The extracted features are then used as inputs for our 1-D CNN that was discussed in the previous chapter. The model is executed at various iterations (epochs) with other variables adjusted in order to optimize the model. The results of these tests are evaluated and discussed in this chapter.

6.1. Data Collection

The objective of our implementation is to have an activity recognition system that uses a small number of sensors that are low cost and can be implemented easily into the daily lives of an elderly person who is beginning to suffer with cognitive decline. We recruited participants through Join Dementia Research, a website developed by the National Institute for Health Research (NIHR), which works in partnership with Alzheimer Scotland, Alzheimer's Research UK, and Alzheimer's Society [187]. The service works closely with healthcare professionals and social care workers to encourage people to participate in Dementia related research studies. Three participants (1 male, 2 female) responded to our invitation to join our study. Each participant was aged over 60 and had either memory loss with no specific diagnosis or had been diagnosed with Mild Cognitive Impairment (MCI).

Individuals with a specific Alzheimer's or Dementia diagnosis were excluded, as we could not guarantee that they had the capacity to consent to participation.

Two Shimmer3 IMU devices were deployed and were worn by participants on the wrist and waist as shown in Figure 6-1 [171]. Sensor position and orientation has been discussed in several studies [143], [152], [163], as we discussed in the sensor investigation. As all of our activities involve some degree of arm movement, we placed a shimmer device on the dominant arm of each participant. An additional device was placed on the waist of participants, as several activities would produce movement through the waist, such as the transitional activities. The sampling rate of the devices was set to 204.8Hz in order to sample sufficient data from each activity, given that some activities can be complex and in part due to the small participant group we possess, high data volume was acceptable.

Accelerometer data and gyroscope data was collected from the Shimmer3 IMU and sent via Bluetooth to a connected Samsung Galaxy S6 smartphone which was running the Multi Shimmer Sync android application. Gyroscopes were activated as an additional sensor as accelerometers alone did not provide sufficient information on an activity to accurately classify it [143]. An additional Samsung Galaxy S6 was used to capture video of participants performing each activity in order to compare the poses and actions a participant makes when performing an activity against the data produced, as this can give us an indication why an activity may be successfully or unsuccessfully classified.

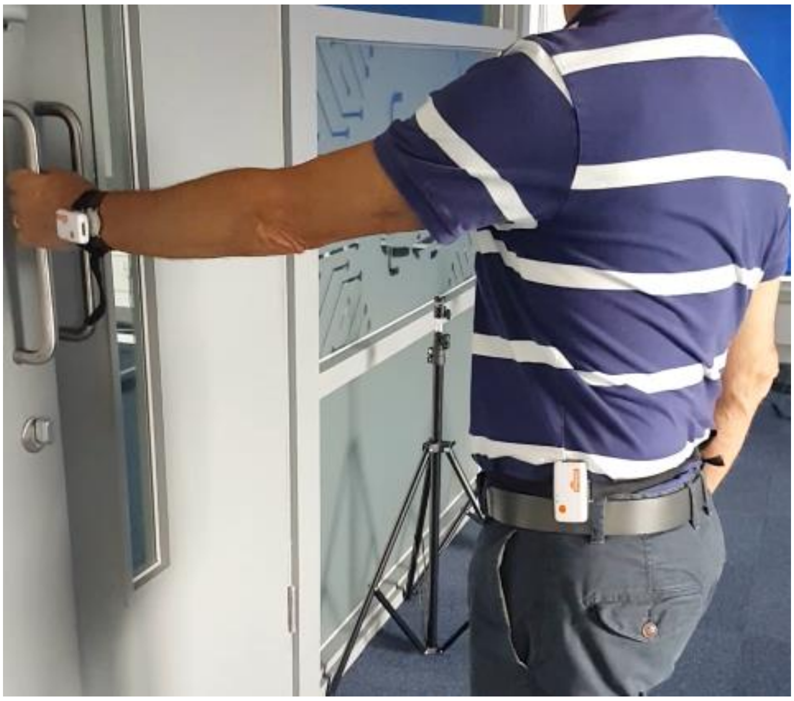


Figure 6-1: Shimmer applied to wrist and waist of participant performing opening door activity

A total of eighteen activities were performed by participants and split into three categories; dynamic, transitional and sedentary as shown in Table 6-1.

Table 6-1: Activities of daily living performed by participants

Activity ID	Activity of Daily Living	Activity Type	Activity Description
A1	Walking	Dynamic	Participant walks up and down a room
A2	Walking Upstairs	Dynamic	Participant walks up a flight of stairs
A3	Walking Downstairs	Dynamic	Participant walks down a flight of stairs
A4	Walking Circle	Dynamic	Participant walks around the room in a circle.

A5	Picking Up Object	Transitional	An object is picked up off a surface and raised to the chest
A6	Placing Object Down	Transitional	An object is placed down onto a surface from the holding positon
A7	Drinking Water	Transitional	A glass of water is raised from the chest to the lips
A8	Opening Door	Transitional	Participant opens a door
A9	Closing Door	Transitional	Participant closes a door
A10	Sitting	Sedentary	Participant remains in an idle sitting position
A11	Standing	Sedentary	Participant remains in an idle standing position
A12	Lying	Sedentary	Participant remains in an idle lying position for
A13	Standing to Sitting	Transitional	Participant goes from a standing to a sitting position
A14	Sitting to Lying	Transitional	Participant goes from a sitting to a lying position
A15	Lying to Sitting	Transitional	Participant goes from a lying to a sitting position
A16	Sitting to Standing	Transitional	Participant goes from a sitting to a standing position
A17	Bending Over to Standing	Transitional	Participant goes from a bent over position to a standing upright position
A18	Standing to Bending Over	Transitional	Participant bends over from a standing upright position

We conducted data collection in a lab-based environment, which was large enough to perform activities such as walking, allowing us to simulate the home-based environment. Additional activities were added over the initial concept in Chapter 4 in order to break up more complicated activities in more simplified steps and reduce any potential noise that could be recorded.

Dynamic activities were performed ten times in a five-second window, with the exception of walking and walking circle which were performed over 1 minute and recorded once and walking up and down stairs which were recorded five times over roughly 10 seconds due to the number of stairs to ascend and descend.

Sedentary activities were performed once by each participant and were recorded over a one-minute duration.

Transitional activities had a window of five seconds and were performed by participants ten times.

Figure 6-2 and Figure 6-3 contains several raw accelerometer signals from the dynamic activity type. Opening door and closing door, display similar patterns to each other, however they are in reverse, particularly on the X and Y axis.

The X axis of opening door does have a flattening of its curve which is a unique feature not shared by the closing door activity.

Placing down object and picking up object are fairly different to one another. Picking up object has a more intense frequency on its X axis while placing object is less intense but does contain some significant peaks.

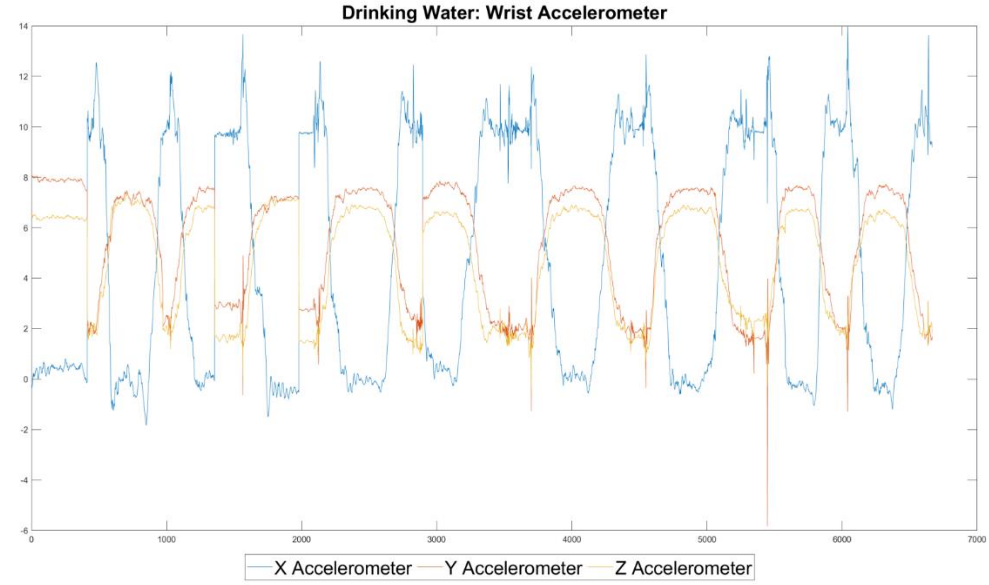
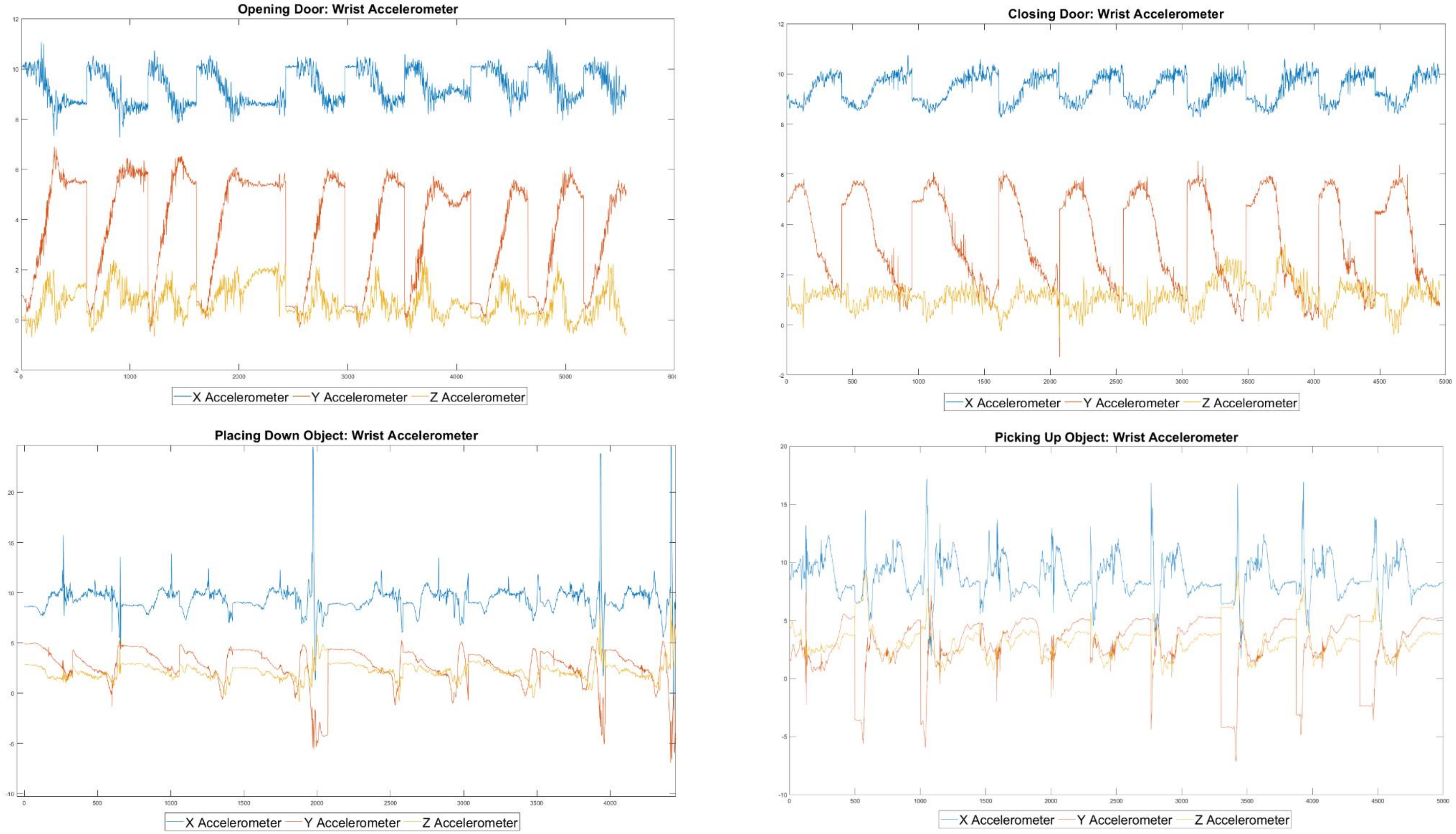


Figure 6-2: Raw Accelerometer Data for the Drinking Water Dynamic Activity


Opening Door: Wrist Accelerometer

Closing Door: Wrist Accelerometer

Placing Down Object: Wrist Accelerometer

Picking Up Object: Wrist Accelerometer

Figure 6-3: Raw accelerometer data for dynamic activities from wrist Shimmer.

Figure 6-4 displays the remaining dynamic activities raw accelerometer data. Walking and walking circle display similar intensity across all three axis and have very similar peaks throughout. However, walking does contain several more significant peaks on all three axes, providing some unique features to aid in classification. Walking up and down stairs has a similar shape in the signal however, walking downstairs contains stronger peaks than walking upstairs.

Figure 6-5 displays raw signals for sedentary accelerometer data. As these activities involve little movement, the signal remains relatively flat. However, lying down does contain some small spikes indicating slight movement of the wrist, while sitting down also contains several curves on the Z axis and several smaller peaks on the X and Y axis, indicating a significant movement of the wrist while the participant was sitting down. Standing has a relatively flat signal for all three axes. This could potentially lead to some misclassification between standing and lying down.

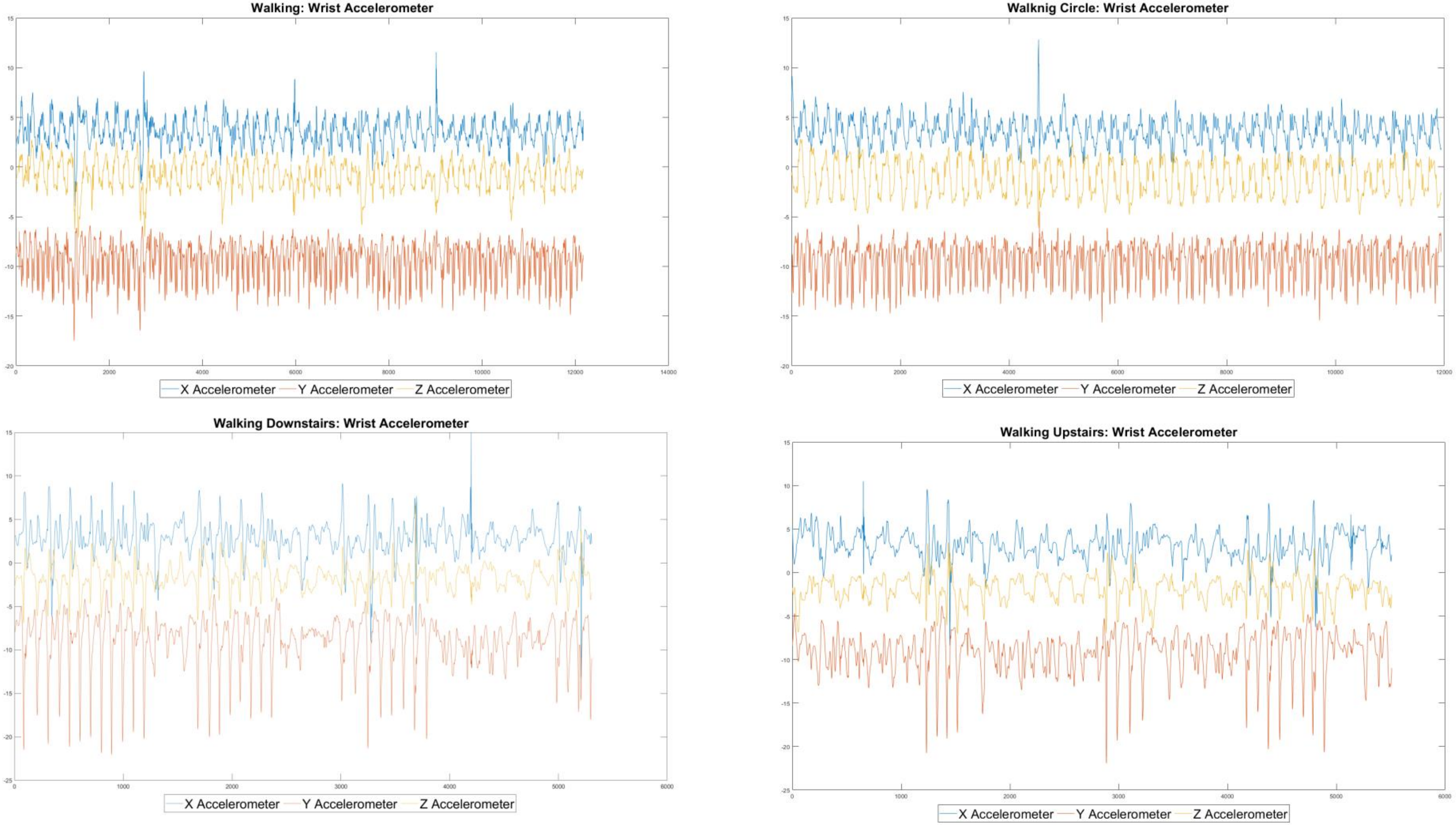


Figure 6-4: Raw accelerometer data for walking activities from wrist Shimmer.

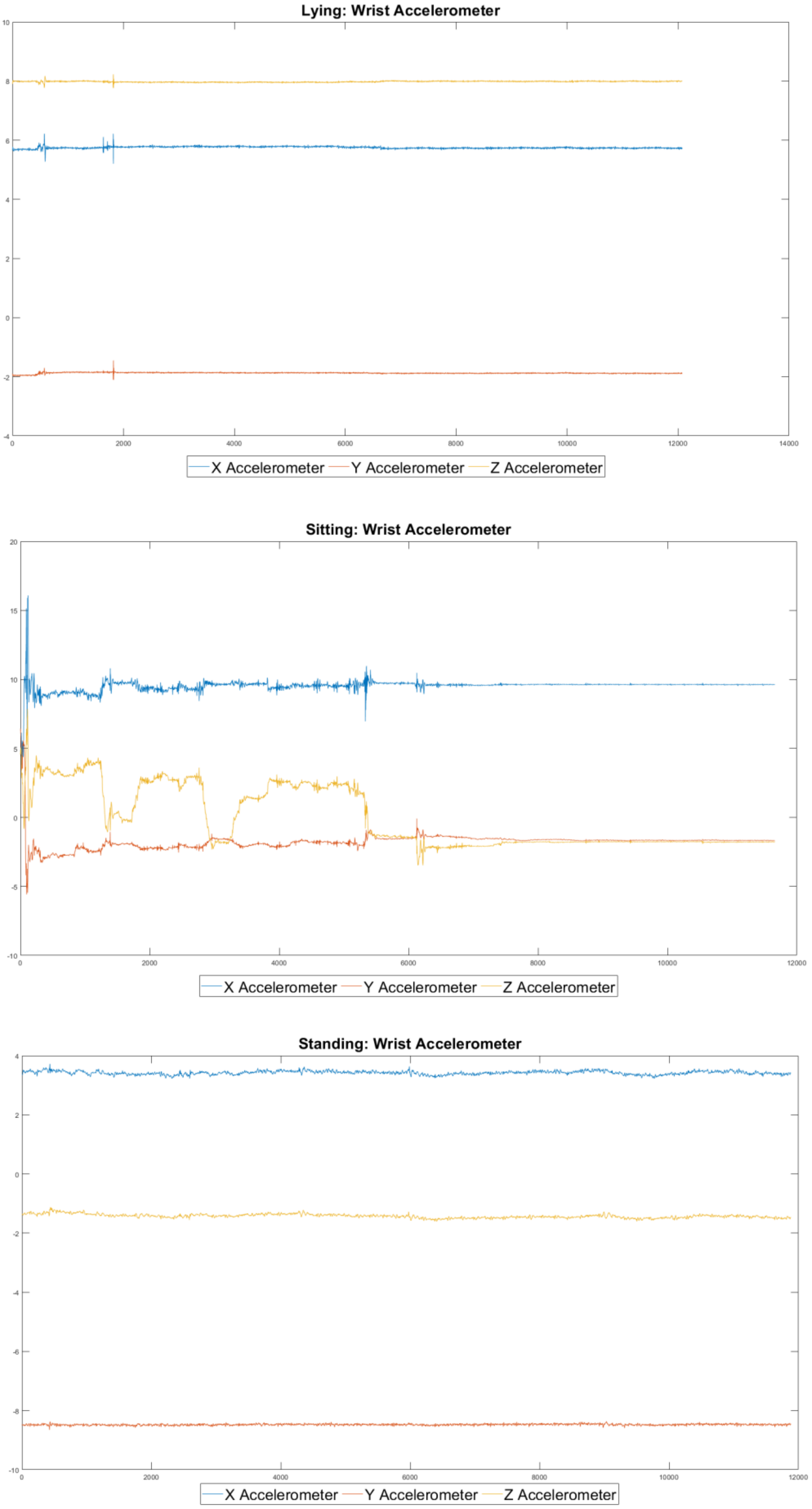


Figure 6-5: Raw accelerometer data for sedentary activities from wrist Shimmer.

Figure 6-6 displays raw signals from the transitional activities performed by participants. Sit-to-lie produces several sharp increases in the peaks on the X and Z axes compared to stand-to-sit which contains small but more frequency peaks. The Y axis of stand-to-sit is also more intense compared to sit-to-lie. Sit-to-stand also produces a number of peaks on the Y axis, however they are not as frequent as stand-to-sit. The Z axis of sit-to-stand does produce larger peaks at several intervals and appears to have more frequent peaks over the activity's duration. Lie-to-sit produces significant intense and frequent peaks for all three axes.

The Y axis could potentially lead to a misclassification with walking upstairs or downstairs due to the similar sharp increase of the peaks. Stand-to-bending over and bend-over-to-stand, produce similar X and Y signal curves, but reversed. This could result in these activities being misclassified as each other. The Z axis of bend-over-to-stand, does produce larger curves compared to stand-to-bend-over, where the Z axis has smaller but more frequent peaks. This difference between the two activities could aid in mitigating potential misclassification.

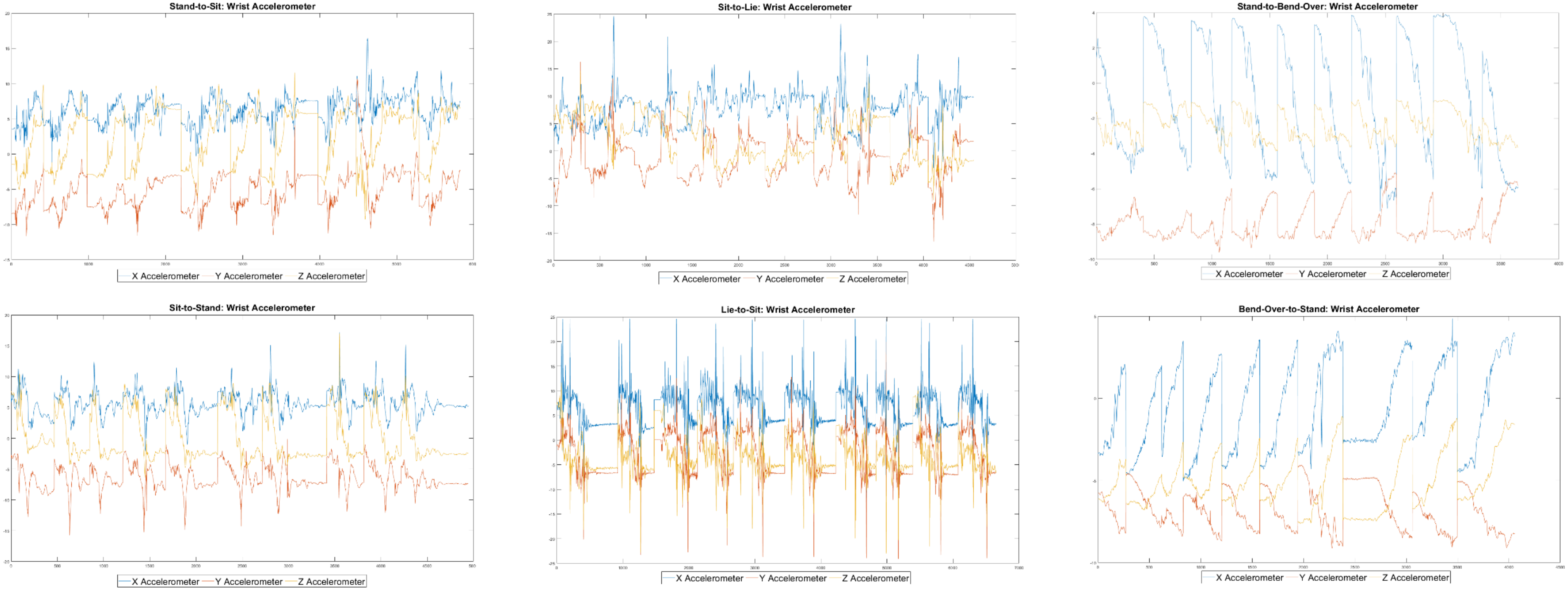


Figure 6-6: Raw accelerometer data from transitional activities from wrist Shimmer.

6.2. Data Pre-Processing

Signal magnitude vector (SMV) is first applied to the data collected from the Shimmer3 devices. SMV is used to evaluate the degree of movement and intensity from acceleration signals where x_i, y_i and z_i represent the axis of the sensor and is displayed in Equation 6-1. It is the most common value for deriving the thresholds of acceleration peaks [177]. A Savitzky-Golay filter is then applied to the data set for the purpose of smoothing the data and to increase the precision of the data [178]. Figure 4-10 provides an example of the application of the filter to walking activity data from the wrist sensor. As the participant numbers were low, restricting the amount of data we could collect, an over-sampling technique known as Smote was applied to the extracted feature data set to increase the total data points. Minority classes are over sampled to create 'synthetic' examples rather than provide an over-sample with replacement [188]. Data sets were also serialized in order to speed up the import time of the data set into the model, thus speeding up the time taken for the model to run.

$$SMV = \sqrt{x_i^2 + y_i^2 + z_i^2}$$

Equation 6-1: Formula for Signal Magnitude Vector

6.3. Feature Extraction

A sliding window of 50% is given to features extracted from the Shimmer accelerometer and gyroscope. Time domain features were extracted from the raw accelerometer and gyroscope data collected from the two Shimmer devices as displayed in

Table 6-2 below. Data sets are standardised and filtered to match the patterns of PA. The results need to be separated due to the multiple sensors consecutively running and collecting data on participant PA. Minimum, Maximum, Average, Correlation, Variance, Standard Deviation, Root Mean Square and Covariance were extracted for the X, Y and Z Axis of the accelerometer and gyroscope. Frequency domain features were also extracted in the form of Fast Fourier Transform (FFT) Energy and Entropy. Standard deviation, variance, correlation and energy have been defined in chapter 4.3. Root Mean Square, covariance and entropy are defined below.

This results in a total of 68 features that are used as inputs to the model as displayed in Figure 5-5 which displays the setup of our 1D Convolutional Neural Network.

Table 6-2: Features extracted from acceleration and gyroscope data

Features Extracted	
Time Domain Features	Mean, Standard Deviation, Min, Max, Correlation, Variance, Covariance, Signal Magnitude Vector, Root Mean Square
Frequency Domain	Fast Fourier Transform Energy, Entropy

Root Mean Square is defined as the square root of the mean values of x_i^2.

$$RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}$$

Equation 6-2: Formula for Root Mean Square

Covariance provides a measure of the correlation strength between two variances (x, y) where the sample size is defined as n.

$$Cov(x, y) = \frac{\sum(x_i - \bar{x})(y_j - \bar{y})}{n - 1}$$

Equation 6-3: Formula for Covariance

Entropy of a signal is a measure of spectral power distribution in the frequency domain. Entropy treats the power distribution as a probability distribution. Entropy can be defined as the following where p_i is the probability distribution.

$$Entropy = -\sum_{i=1}^{n} p_i(log_2 p_i)$$

Equation 6-4: Formula for Entropy

6.4. Classification Results and Discussion

Extracted features were used as inputs to the CNN that we discussed in chapter 5.5. Several iterations of the model were tested using ReLu, Sigmoid and Tanh activation functions with Softmax applied to the final layer of the model.

Each model was tested for 100, 200, 500 or 1,000 epochs in order to gauge the most appropriate epoch for the model's learning and ensuring the model can maintain its ability to generalise.

Additional iterations explored changing the final layer of the model from using Softmax to using the designated activation function on all four layers. Batch sizes were also changed from 64 to 32 and 128 to monitor any potential impact the batch size had on the overall accuracy. These additional iterations were performed at 200 epochs for all three activation functions.

A computational rate of 200 epochs was chosen for these iterations of the model, in order to ensure the model had sufficient time to learn and display any underfitting or overfitting.

Further results from our tests can be seen in Appendices C – P. These contain additional graphs and charts further displaying the performance of the model and confusion matrices displayed the accuracy rate for each activity.

6.4.1. Evaluation of the CNN model at multiple Epochs

Total accuracies of the CNN classification of both wrist and waist Shimmer data can be seen in Table 6-3. ReLu managed to achieve the highest accuracies in most instances over the other two activation functions with the exception of Sigmoid at 1000 epochs, which achieved 86% accuracy on wrist data over ReLu. Tanh had the worst performance as it struggled to classify above 80% in several instances, with its lowest accuracy at 77% for 1,000 epochs. Each activation had an epoch where it performed best. For ReLu, it was 200 epochs for the wrist, and 1,000 for the waist. However, the wrist data set had an accuracy of 85% at 1,000 epochs, 1% less than 200. The gains made to the accuracy of the waist data set may not be worth the additional computation time over 2,000 epochs, however, they could be justified as significant over training at 100 epochs. Sigmoids best performance was at 1,000 epochs, where it achieved 86% for the wrist data set and 85% for the waist. This is a similar result to ReLu at 200 epochs. The additional time taken for Sigmoid to achieve the same result as ReLu is not justified. Tanh's best result was at 500 epochs where it managed to achieve over 80% on both data sets. However, this is still lower than the lowest accuracy from ReLu at 100 epochs, and also lower than Sigmoid at several instances.

As a result, Tanh may to be the optimal activation function to use for the classification of our data sets.

Table 6-3: Total Accuracy (%) of each model evaluation produced using CNN.

Accuracy %						
	ReLu		Sigmoid		Tanh	
Epoch	Wrist	Waist	Wrist	Waist	Wrist	Waist
100	0.831884	0.846957	0.818551	0.803478	0.806377	0.797681
200	0.862609	0.858551	0.81913	0.826667	0.794203	0.811594
500	0.846377	0.864928	0.845217	0.845217	0.811594	0.826667
1000	0.854493	0.88	0.861449	0.857391	0.777971	0.813913

The loss for each iteration of the CNN model can be seen in Table 6-4. Loss is a summation of the errors that are encountered during classification when the predicted output has deviated from the target output.

Therefore, a lower loss produced by the CNN model means that the model performs better and is better optimized. Tanh can be seen to have a higher loss over ReLu and Sigmoid, achieving a loss of 0.75 at 1,000 epochs. Sigmoid also had high losses over ReLu at each epoch, a loss of 0.56 and 0.54 at 200 epochs over ReLu 0.48 and 0.47. As it appears to have the lowest loss, ReLu would be the preferred activation function. ReLu achieves a higher loss on the wrist data at 1,000 epochs and has its lowest loss at 100 epochs on the waist data set. The lowest loss on the wrist data set is at 200 epochs achieving 0.48 and the waist loss is 0.47 which is marginally higher than the loss at 100 epochs. As a result, we believe that 200 epochs is a sufficient amount of training time for our data set to achieve a good accuracy and relatively low loss.

Table 6-4: Total Loss of each model evaluation produced using CNN.

Loss						
	ReLu		Sigmoid		Tanh	
Epoch	Wrist	Waist	Wrist	Waist	Wrist	Waist
100	0.505462	0.429599	0.585236	0.589009	0.655279	0.659082
200	0.477348	0.468198	0.561673	0.536886	0.654045	0.606578

500	0.535944	0.477318	0.508893	0.503796	0.627709	0.569655
1000	0.663337	0.487609	0.49627	0.459829	0.754157	0.61493

a) CNN Model Training at 100 Epochs

Figure 6-7 displays the training and validation accuracy over the course of 100 epochs for ReLu, Sigmoid and Tanh on the wrist data set. All three have a consistent curve showing the increase in accuracy as training occurs with each achieving a high accuracy by 100 epochs. ReLu does manage to achieve a training accuracy at about 85%, this result was closely obtained during testing with a score of 83% as seen in Table 6-3. Sigmoid and Tanh achieve a training accuracy above 80%, and still achieve close to 80% in testing. The curves shown on the training set clearly indicate the model is learning. We can also see the validation result for Sigmoid and Tanh converge with the training set, for ReLu the validation increases to above 90%, indicating that the model has a good capability to generalize data.

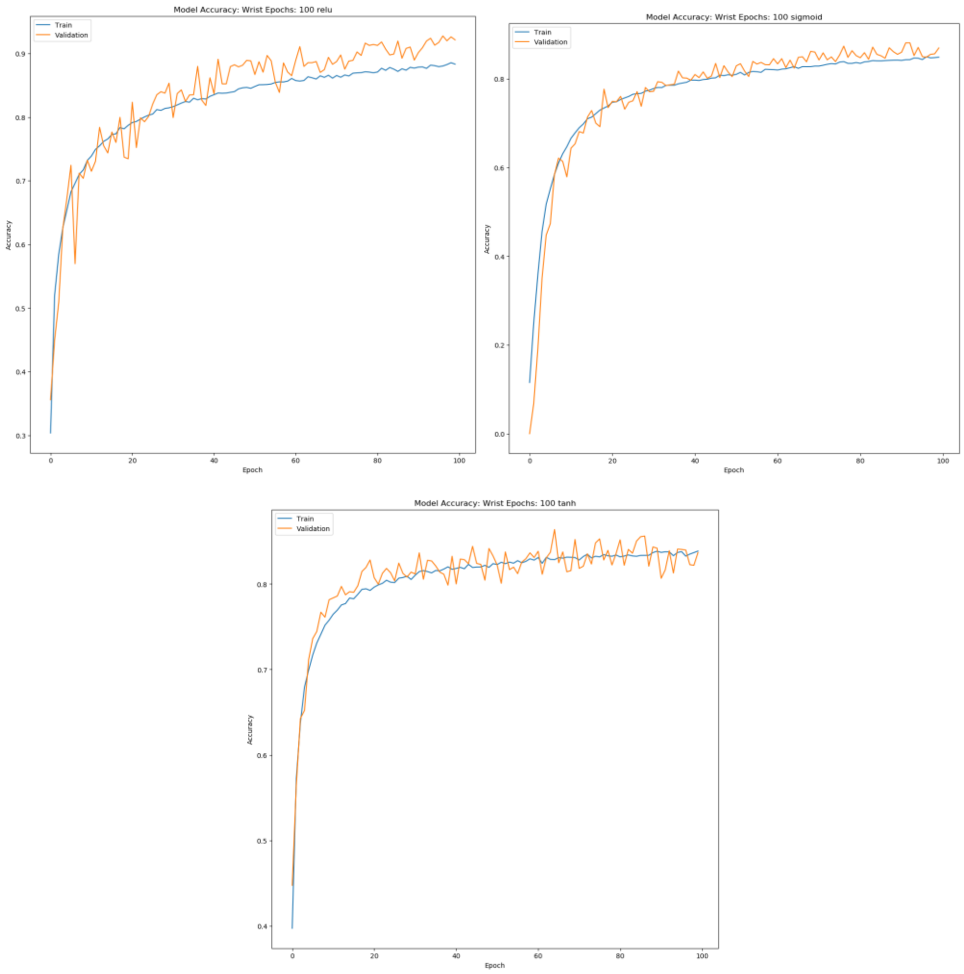


Figure 6-7: Training and Validation Accuracy at 100 Epochs for Wrist Data set.

From Top-Left: ReLu, Sigmoid and Tanh

Figure 6-8 displays the training and validation loss for the wrist data set at 100 epochs. During training we can see that the loss is constantly decreasing, producing a relatively flat curve, supporting our earlier remark about the model's learning during training. Validation for ReLu and Sigmoid is also decreasing constantly, indicating the model's ability to generalise. However, for Tanh the validation loss begins to increase which means that the model here is beginning to overfit the training data. This can also be observed in Figure 6-9 which displays the histogram of kernel weights for the Tanh model. The shape of the distribution curve on each layer begins to remain the same as the model approaches 100 epochs. This can mean that the model has stopped learning, which we have seen from the accuracy and loss that it is learning, or it can mean that the model is overfitting the data. This could potentially be the reason why Tanh does not achieve the same accuracy as ReLu or Sigmoid for our model.

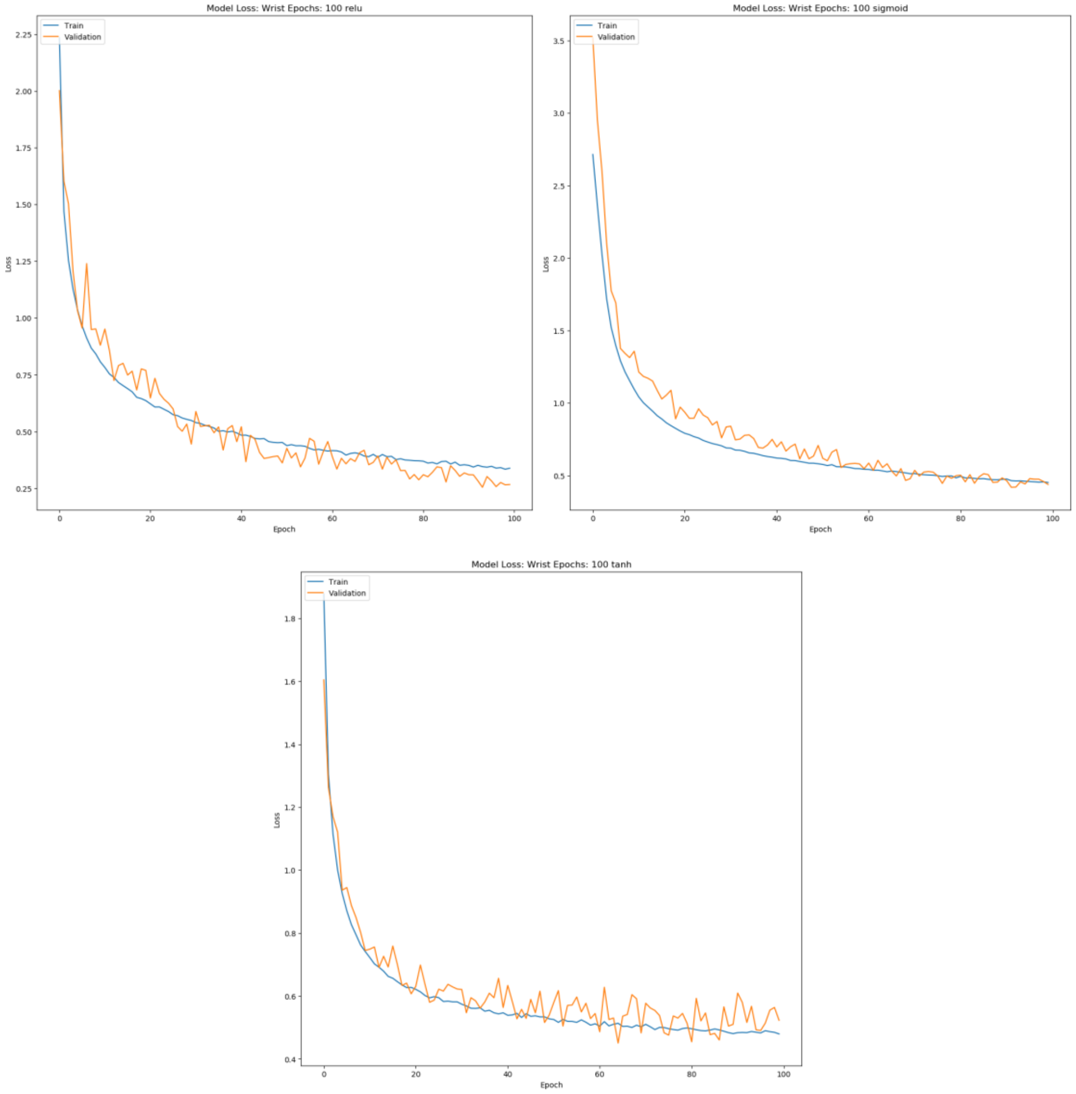


Figure 6-8: Training and Validation Loss at 100 Epochs for Wrist Data set.

From Top-Left: ReLu, Sigmoid and Tanh

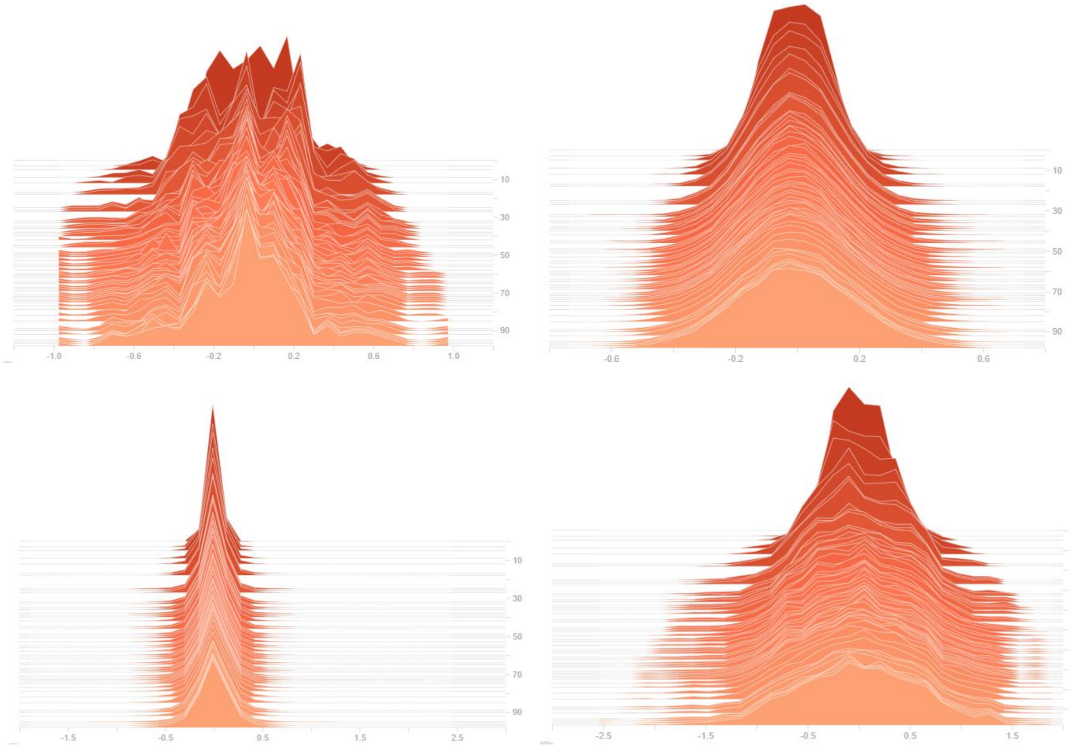


Figure 6-9: Wrist data set Histogram of kernel weights for the four layers of the network using Tanh at 100 epochs.

From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.

b) CNN Model Training at 200 Epochs

The wrist and waist data set's training and validation accuracy for activations functions at 200 epochs can be seen in Figure 6-10.

ReLu manages to achieve above 90% in training and validation, maintaining its ability to learn and to generalize the data set however, improving upon the results shown at 100 epochs above. Sigmoid remains roughly the same at 200 epochs for training and validation compared to 100 epochs. This can also be seen in the testing accuracy results in Table 6-3. Tanh's training accuracy remains roughly the same, however validation accuracy does show better generalization ability compared to 100 epochs.

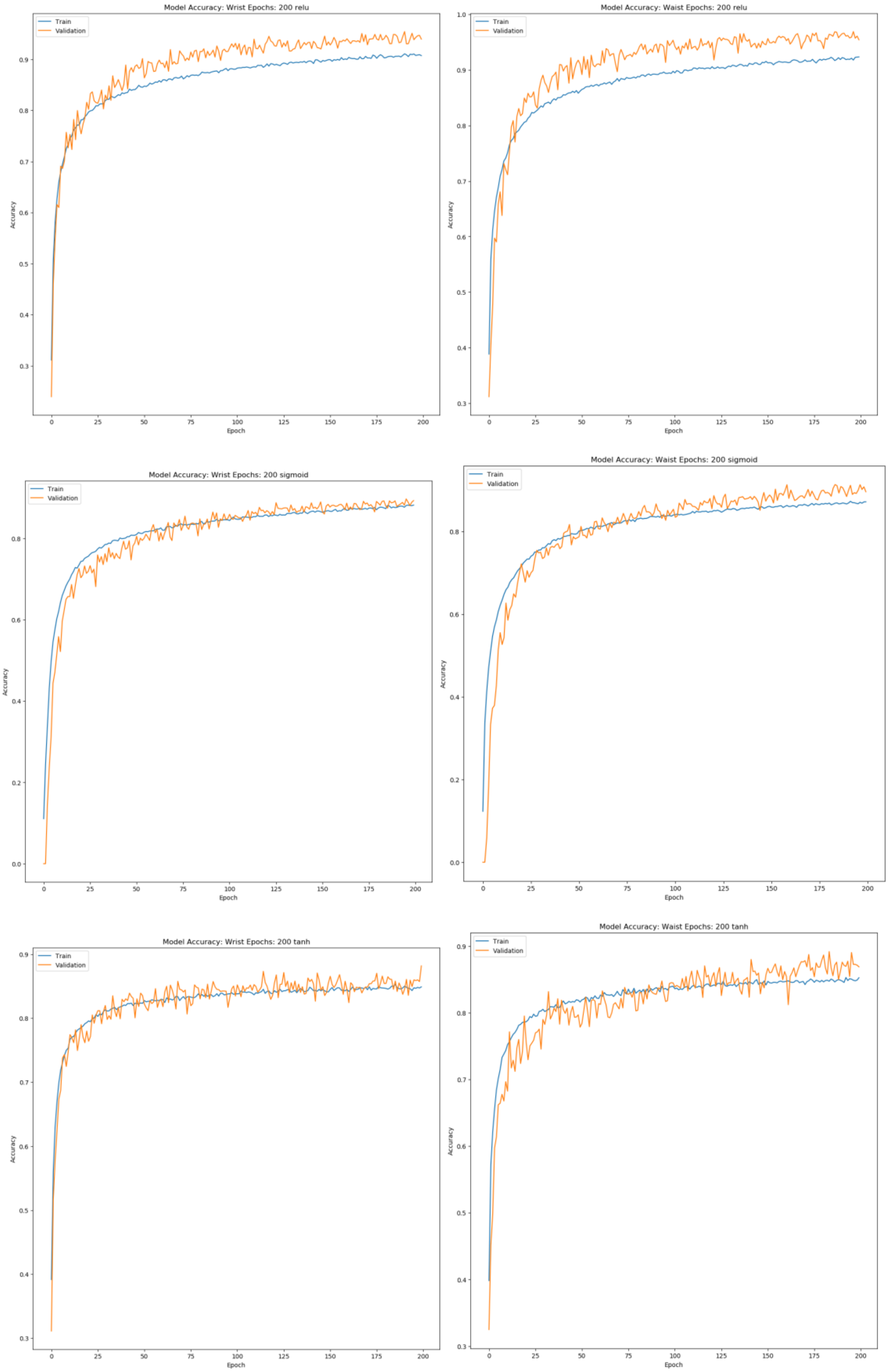


Figure 6-10: Training and Validation Accuracy at 200 Epochs for Wrist and Waist Data set.

From Top: ReLu, Sigmoid and Tanh

Figure 6-11 displays the loss of the model at 200 epochs using the Tanh function. Here we can see the training loss begin to flatten out as it approaches 200 epochs. Learning does appear to still be occurring, as shown with the accuracy above however, the loss would suggest this is slowing down. The validation loss does not begin to increase again as it did with 100 epochs. The validation loss converges on the training loss and even begins to dip below training on the waist set. This suggests that 200 epochs allows the model, using Tanh, to generalize the data set better compared to 100 epochs. This does display an improvement in accuracy for the waist data set but does lead to a small accuracy decrease on the wrist data set.

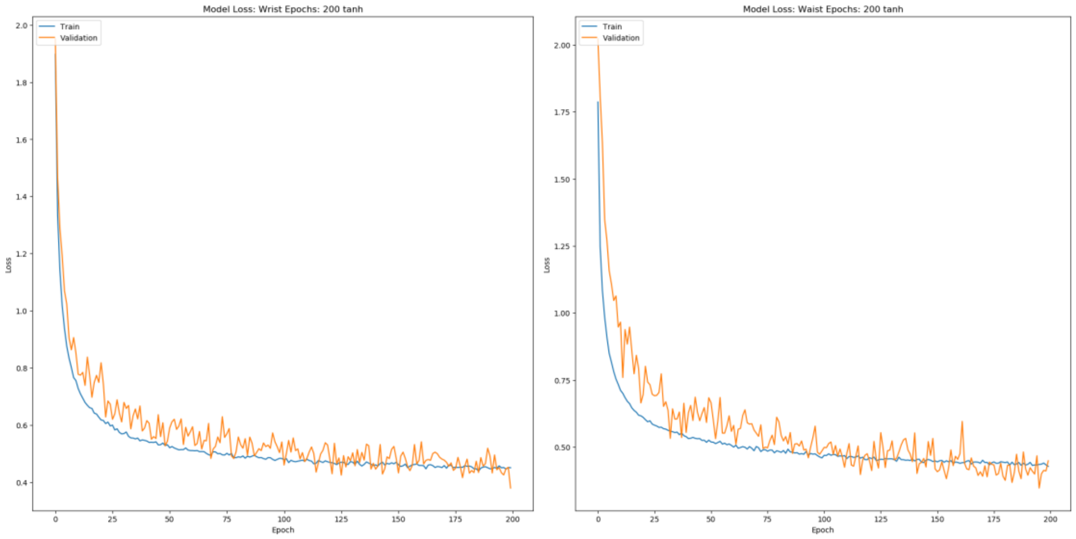


Figure 6-11: Training and Validation Loss at 200 Epochs with Tanh function for Wrist and Waist Data set

Figure 6-12 and Figure 6-13 display the confusion matrices for the wrist and waist data sets for ReLu at 200 epochs. Sedentary activities achieved the highest positive classification on both data sets. Sitting was misclassified as walking upstairs on both data sets. It was also misclassified as the transitional activities sit-to-lie and lie-to-sit on the wrist data set, and sit-to-stand and stand-to-sit on the waist data set. Lying was able to achieve a 100% classification rate on both data sets.

The dynamic activity, walking, achieved an 81% and 78% accuracy on both data sets, with the activity mainly being misclassified as walking circle on the wrist data set and walking upstairs on the waist set. Walking upstairs only achieved 67% on the wrist data set with most misclassifications occurring on lie-to-sit and walking downstairs. Opening door and closing door on the wrist data set achieved relatively high accuracies, with the majority of opening door misclassifications being closing door and vice versa. A similar pattern occurs for closing door on the waist data set however, opening door struggles to achieve a good accuracy with it mainly being misclassified as walking up and down stairs.

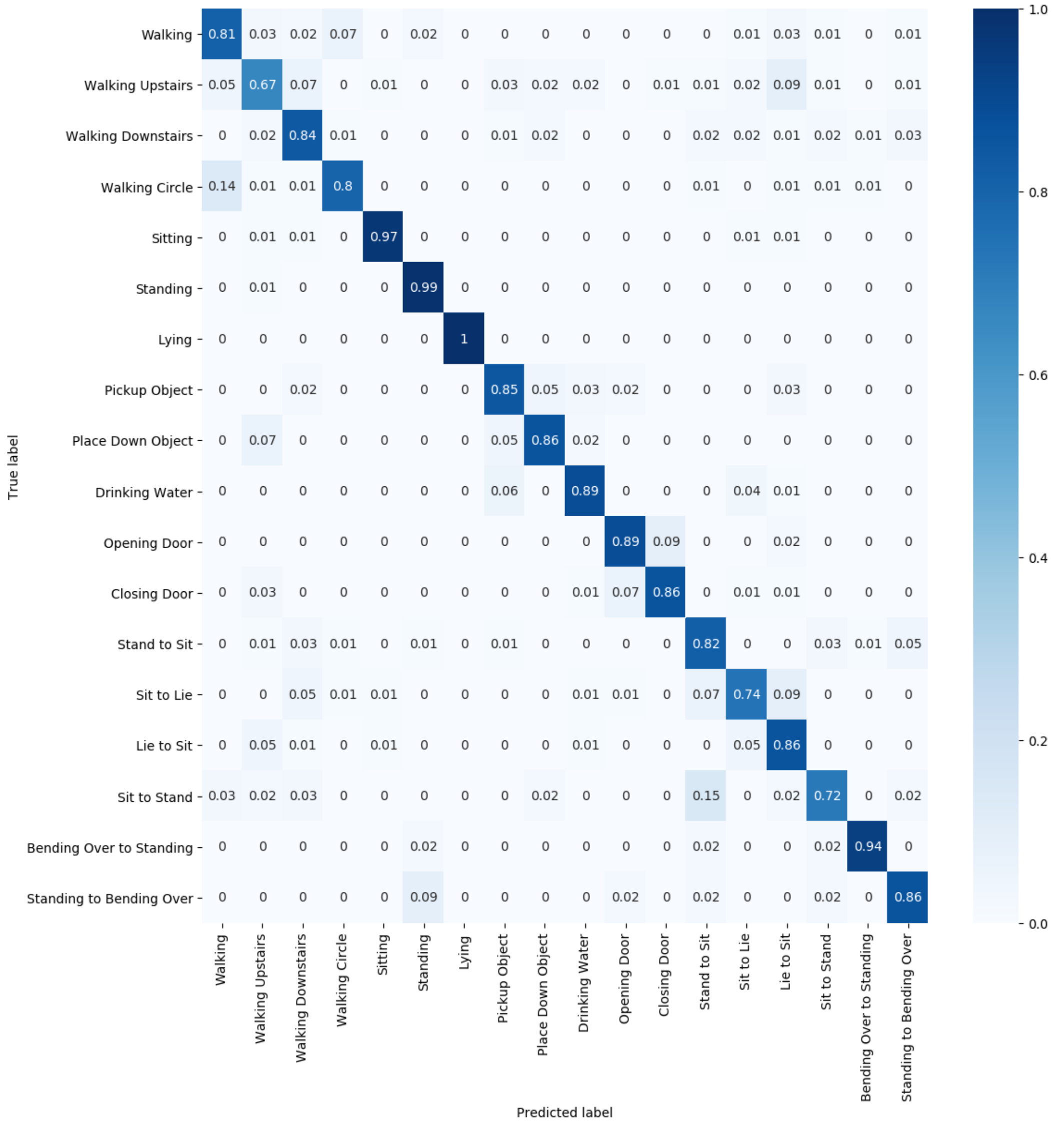
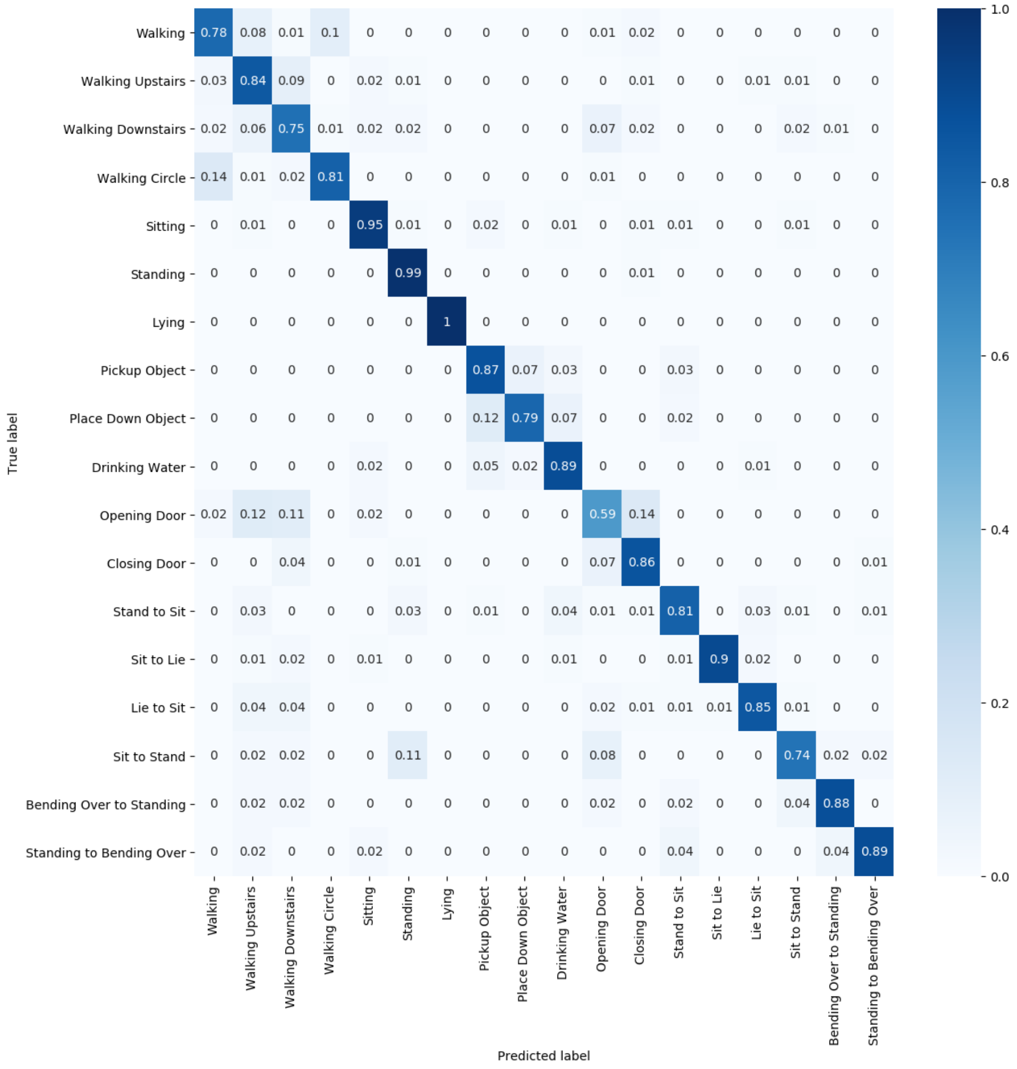


Figure 6-12: Confusion matrix for Wrist Data using ReLu at 200 Epochs

Transitional activities achieved relatively good accuracies on both data sets, with most achieving above 80%. Bend-to-stand and stand-to-bend achieved good scores, while being misclassified as standing on the wrist data set possibly due to the standing aspect of the activity. While on the waist data set stand to bend was mainly misclassified as bend-to-stand as well as stand-to-sit. Sit-to-stand achieved the lowest accuracy of the transitional activities on the wrist data set, scoring 72% with it being mostly misclassified as stand-to-sit. This will most likely be due to the standing aspects of the activity. It also achieved the lowest accuracy for transitional activities on the waist data set scoring 74% with it mainly being misclassified as standing and opening door. This will also be due to the standing natures of these activities.



	Walking	Walking Upstairs	Walking Downstairs	Walking Circle	Sitting	Standing	Lying	Pickup Object	Place Down Object	Drinking Water	Opening Door	Closing Door	Stand to Sit	Sit to Lie	Lie to Sit	Sit to Stand	Bending Over to Standing	Standing to Bending Over
Walking	0.78	0.08	0.01	0.1	0	0	0	0	0	0	0.01	0.02	0	0	0	0	0	0
Walking Upstairs	0.03	0.84	0.09	0	0.02	0.01	0	0	0	0	0	0.01	0	0	0.01	0.01	0	0
Walking Downstairs	0.02	0.06	0.75	0.01	0.02	0.02	0	0	0	0	0.07	0.02	0	0	0	0.02	0.01	0
Walking Circle	0.14	0.01	0.02	0.81	0	0	0	0	0	0	0.01	0	0	0	0	0	0	0
Sitting	0	0.01	0	0	0.95	0.01	0	0.02	0	0.01	0	0.01	0.01	0	0	0.01	0	0
Standing	0	0	0	0	0	0.99	0	0	0	0	0	0.01	0	0	0	0	0	0
Lying	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Pickup Object	0	0	0	0	0	0	0	0.87	0.07	0.03	0	0	0.03	0	0	0	0	0
Place Down Object	0	0	0	0	0	0	0	0.12	0.79	0.07	0	0	0.02	0	0	0	0	0
Drinking Water	0	0	0	0	0.02	0	0	0.05	0.02	0.89	0	0	0	0	0.01	0	0	0
Opening Door	0.02	0.12	0.11	0	0.02	0	0	0	0	0	0.59	0.14	0	0	0	0	0	0
Closing Door	0	0	0.04	0	0	0.01	0	0	0	0	0.07	0.86	0	0	0	0	0	0.01
Stand to Sit	0	0.03	0	0	0	0.03	0	0.01	0	0.04	0.01	0.01	0.81	0	0.03	0.01	0	0.01
Sit to Lie	0	0.01	0.02	0	0.01	0	0	0	0	0.01	0	0	0.01	0.9	0.02	0	0	0
Lie to Sit	0	0.04	0.04	0	0	0	0	0	0	0	0.02	0.01	0.01	0.01	0.85	0.01	0	0
Sit to Stand	0	0.02	0.02	0	0	0.11	0	0	0	0	0.08	0	0	0	0	0.74	0.02	0.02
Bending Over to Standing	0	0.02	0.02	0	0	0	0	0	0	0	0.02	0	0.02	0	0	0.04	0.88	0
Standing to Bending Over	0	0.02	0	0	0.02	0	0	0	0	0	0	0	0.04	0	0	0	0.04	0.89

Figure 6-13: Confusion matrix for Waist Data using ReLu at 200 Epochs

A graph of statistics for the wrist data set can be seen in Figure 6-14.

We can see that there is a high sensitivity compared to a low false negative rate (FNR) for many activities such as lying, bend-over-to-stand and opening door. We can also see a high specificity compared to a low false positive rate (FPR) for the majority of the activities. We can also observe a high probability of positive outputs and low probability of negative outputs with precision and the negative predictive value. There is also a low false discovery rate (FDR) for most activities meaning we can predict that at most 27% of positive results will be truly negative. We can also observe the lowest f1 score of 68% for walking downstairs up to 100% for lying down. We can also see a relatively high MCC value for each activity, meaning better predictions for activities on the wrist data set.

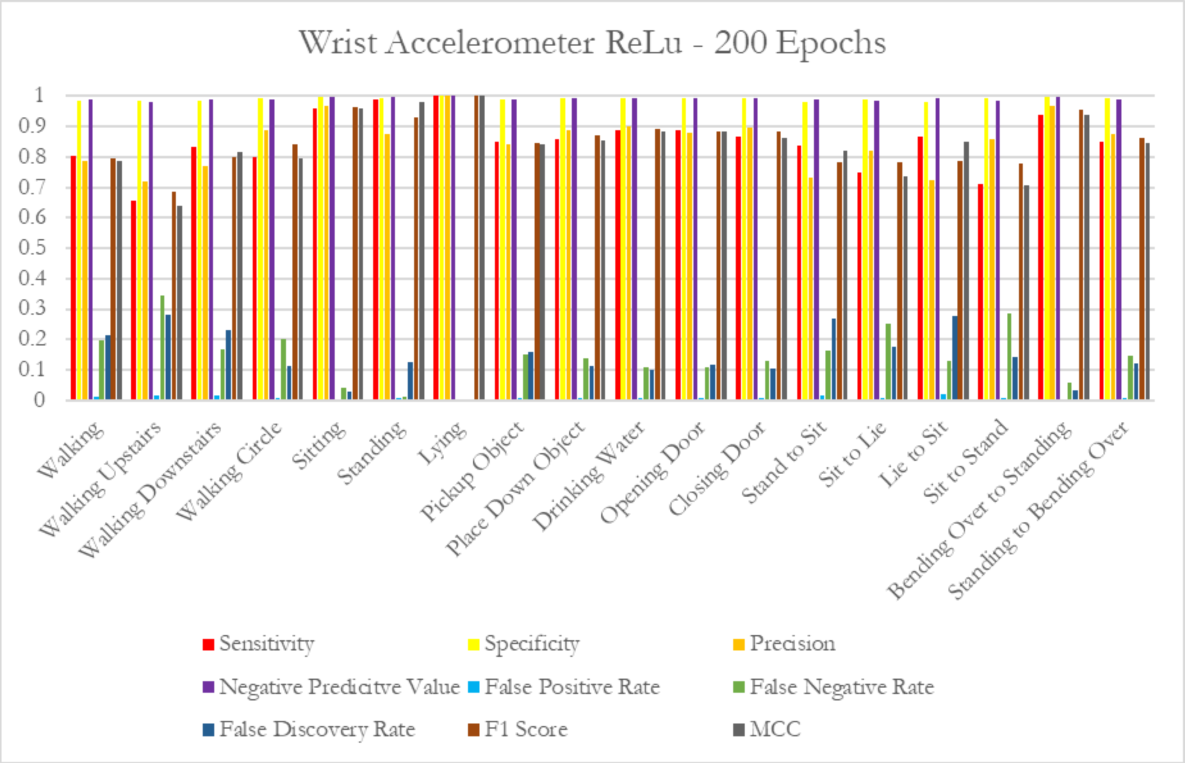


Figure 6-14: CNN Statistics for Wrist Data set on ReLu at 200 Epochs

A graph of statistics for the waist data set can be seen in Figure 6-15. We can observe the low sensitivity of opening door at 59% compared to its FNR of 41%. Other sensitivity values remain fairly high compared to low FNR values. Specificity is also high compared to the FPR for each activity, similar to the wrist data set. The precision for walking up and downstairs is fairly average, both at 66% probability of being identified positive while the negative predictive value is high for these activities.

Walking up and down stairs also achieve a 33% FDR, meaning out of all data there is roughly a 33% probability of a negative being identified positive.

Walking downstairs also achieves the lowest f1 score with walking upstairs being only slightly more accurate. Other activities, such as sit-to-lie achieve high f1 scores. MCC also displays high results for each activity meaning that have achieved a high positive correlation.

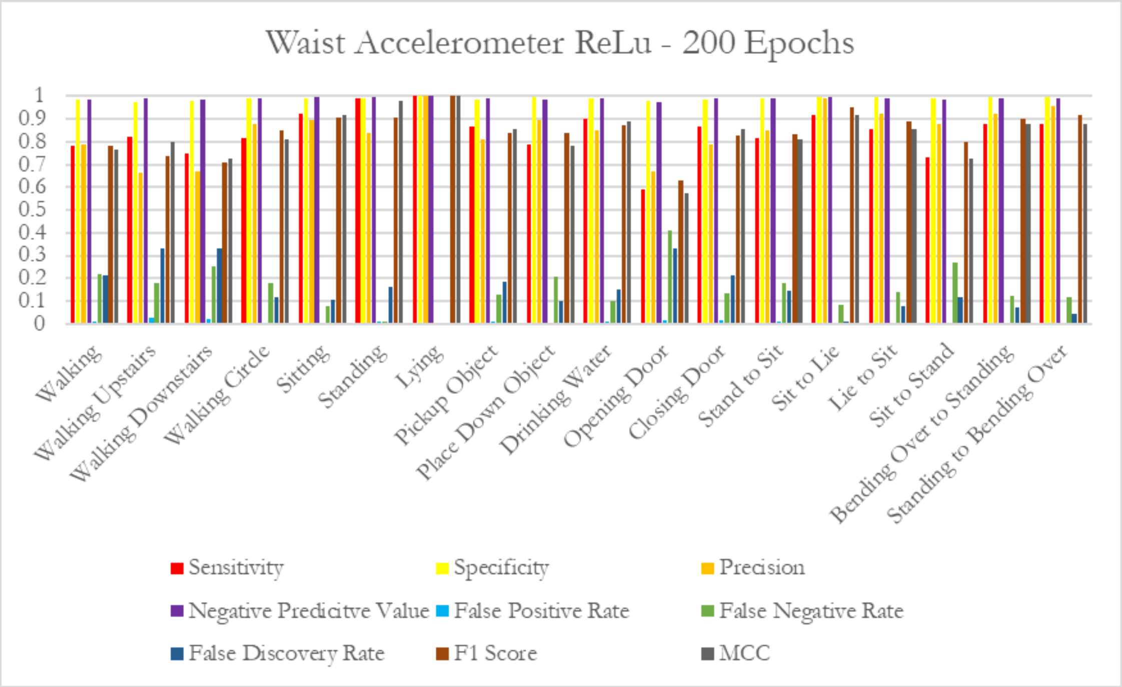


Figure 6-15: CNN Statistics for Waist Data set on ReLu at 200 Epochs

The learning of the neural network can be observed through histograms of the weights for the wrist data set as seen in Figure 6-16 and Figure 6-17. We can see the layer 1 conv1D in Figure 6-16 the weights and distribution of the weights change quite dramatically as each epoch (Y-axis) occurs. This indicates that the network is constantly learning and adapting as it is being trained. The dense layer 2 can also be seen to be constantly learning over time, although the changes to the weights are not quite as dramatic. As this is the final layer with the objective of achieving the final output, we can see that only slight adjustments are being made to the weight distribution in order to ensure we achieve the correct classification.

The conv1D layer 2 and dense layer 1 do not appear to be changing over time after the initial few epochs, maintaining the same shape for the weight distribution. This could be indicative of the network not learning during these layers, potentially leading to an overfitting issue.

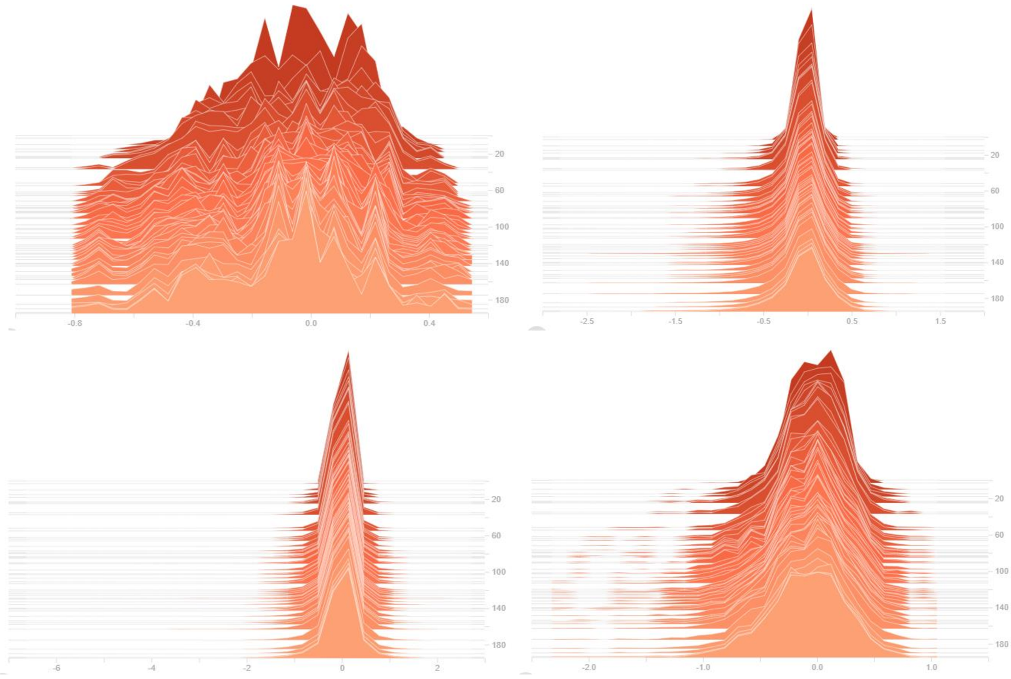


Figure 6-16: Wrist data set Histogram of kernel weights for the four layers of the network using ReLu at 200 epochs.

From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.

For the bias weights in Figure 6-17 we can observe dramatic changes in weight distribution for the first few epochs. After the initial few epochs, the changes in distribution appear much smaller, however they do appear to be constantly changing. This is indicative of the network constantly learning over time for all four layers meaning the network will be able to generalize well.

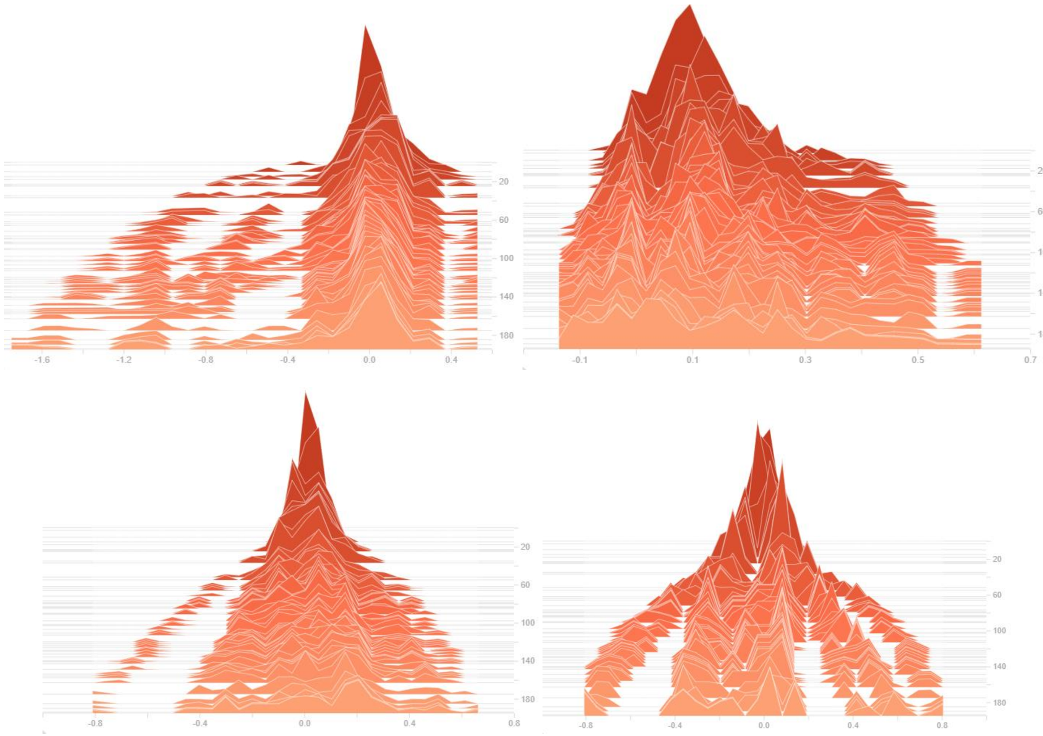


Figure 6-17: Wrist data set Histogram of bias weights for the four layers of the network using ReLu at 200 epochs.

From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.

This can also be observed with the distribution graphs in Figure 6-18 where we can also see the changes to bias distribution over time, where the darker coloured zones indicated more bias weights sitting in that range.

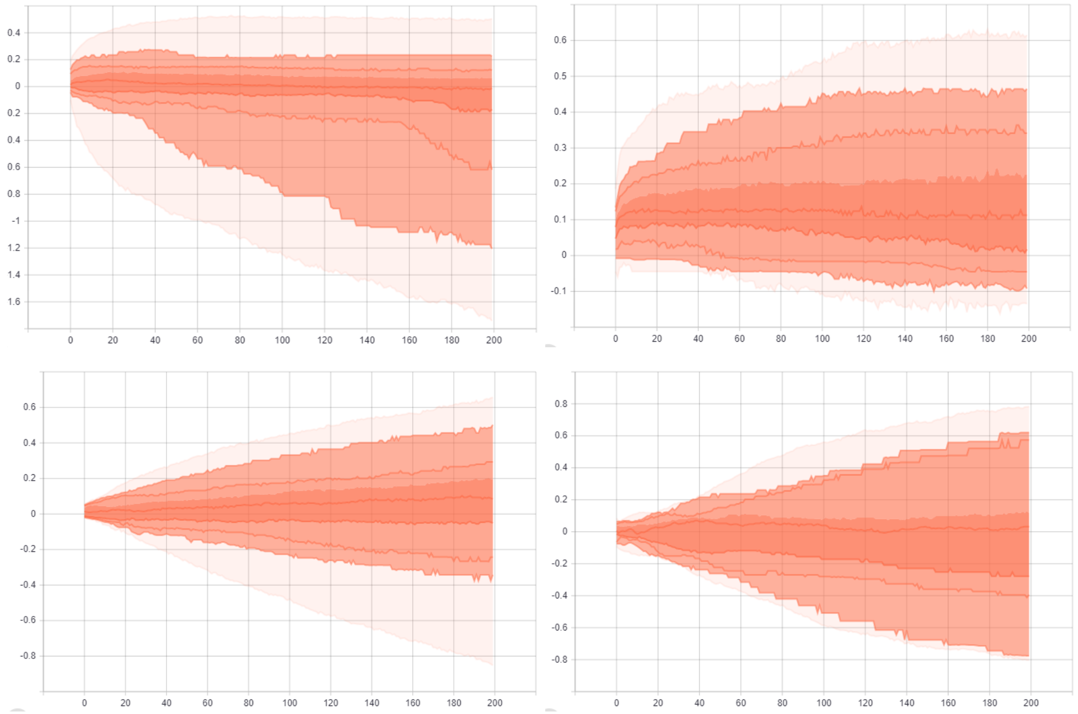


Figure 6-18: Wrist data set Distribution of bias weights for the four layers of the network using ReLu at 200 epochs.

From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.

The kernel weights for the waist data as displayed in Figure 6-19 also display a constant change in weight distribution over time. The conv1D layer 2 does mostly maintain a similar shape during training, with only small changes occurring to the distribution of weight values below -1, indicating that learning is still occurring despite it being small and subtle. The remaining graphs do display the similar patterns observed above, with significant changes occurring during the first few epochs, and smaller adjustments being made to weight distribution as time training goes on. Further graphs for the CNN model using ReLu at 200 epochs can be viewed under Appendix D.

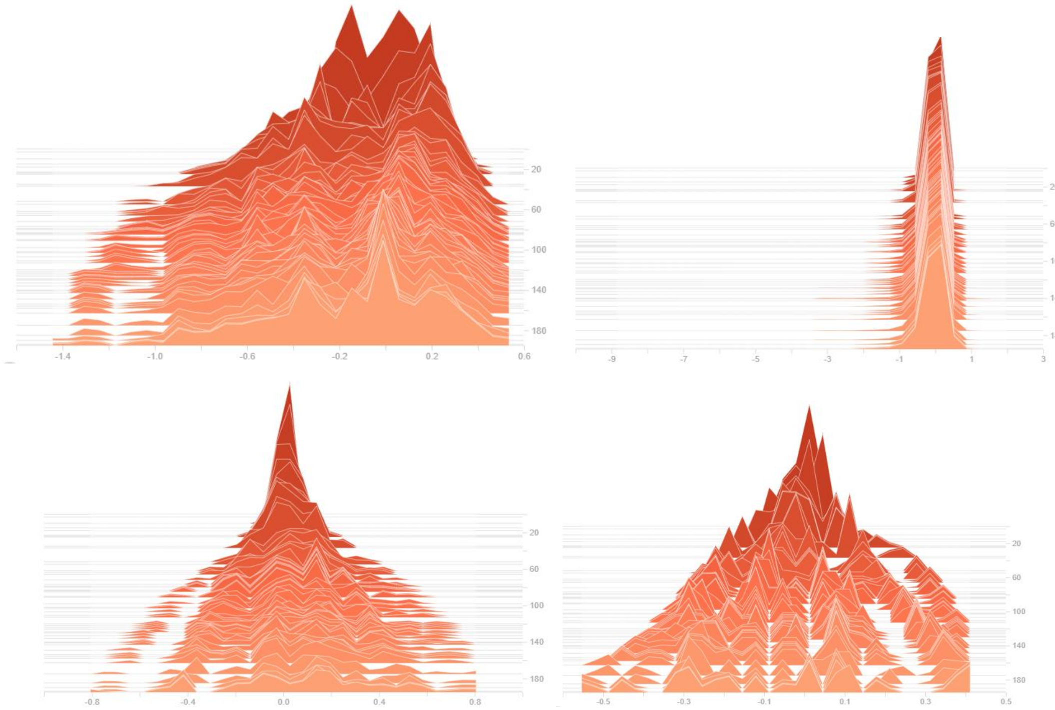


Figure 6-19: Waist data set Distribution of kernel weights for the four layers of the network using ReLu at 200 epochs.

From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.

c) CNN Model Training at 500 Epochs

The accuracy of the waist data set at 500 epochs is shown in Figure 6-20. Training accuracy doesn't generally improve for the activation functions at this epoch compared to 200 epochs in Figure 6-10. ReLu manages to achieve an accuracy over 90% while Sigmoid and Tanh achieve over 85%, and only small gains in testing accuracy are achieved over 200 epochs as shown in Table 6-3. The additional computation time for the model to achieve such a small increase would not be justified in a longer-term study, where speed would be beneficial. Additional data from participants would be the solution to improve the accuracy further.

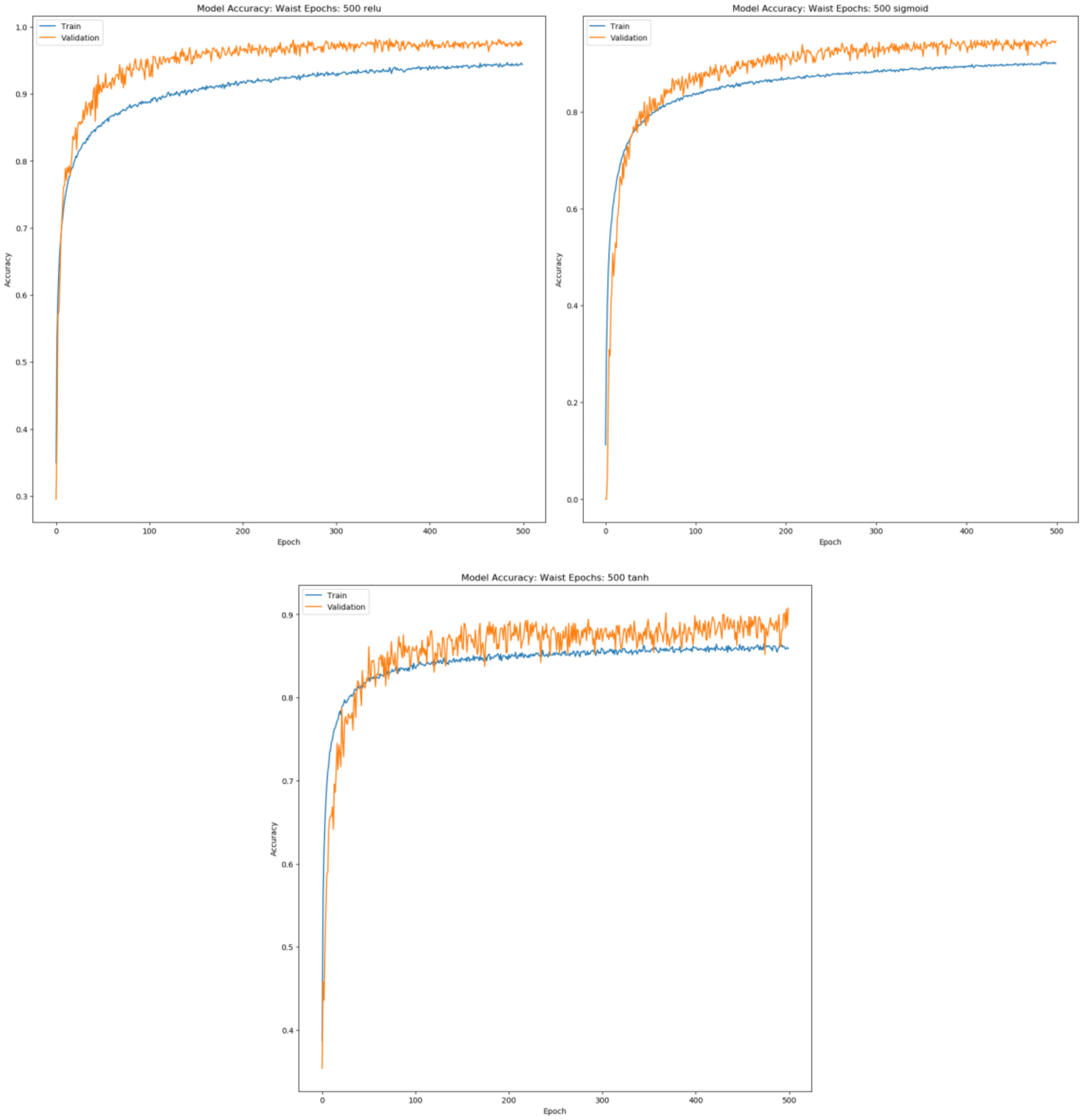


Figure 6-20: Training and Validation Accuracy at 500 Epochs function for Waist Data set.

From top-left: ReLu, Sigmoid, Tanh.

Training loss, displayed in Figure 6-21, does show improvement over 200 epochs as seen in Figure 6-11 as well as appendix D, J and N. The improvements are small but do indicate that learning is still occurring. Validation remains roughly the same for ReLu and Sigmoid, however Tanh does show an improvement over what was displayed in Figure 6-11 meaning its ability to generalize does still improve over 200 epochs. This can also be seen in Table 6-4 where the testing loss at 500 epochs is lower than 100 and 200 epochs.

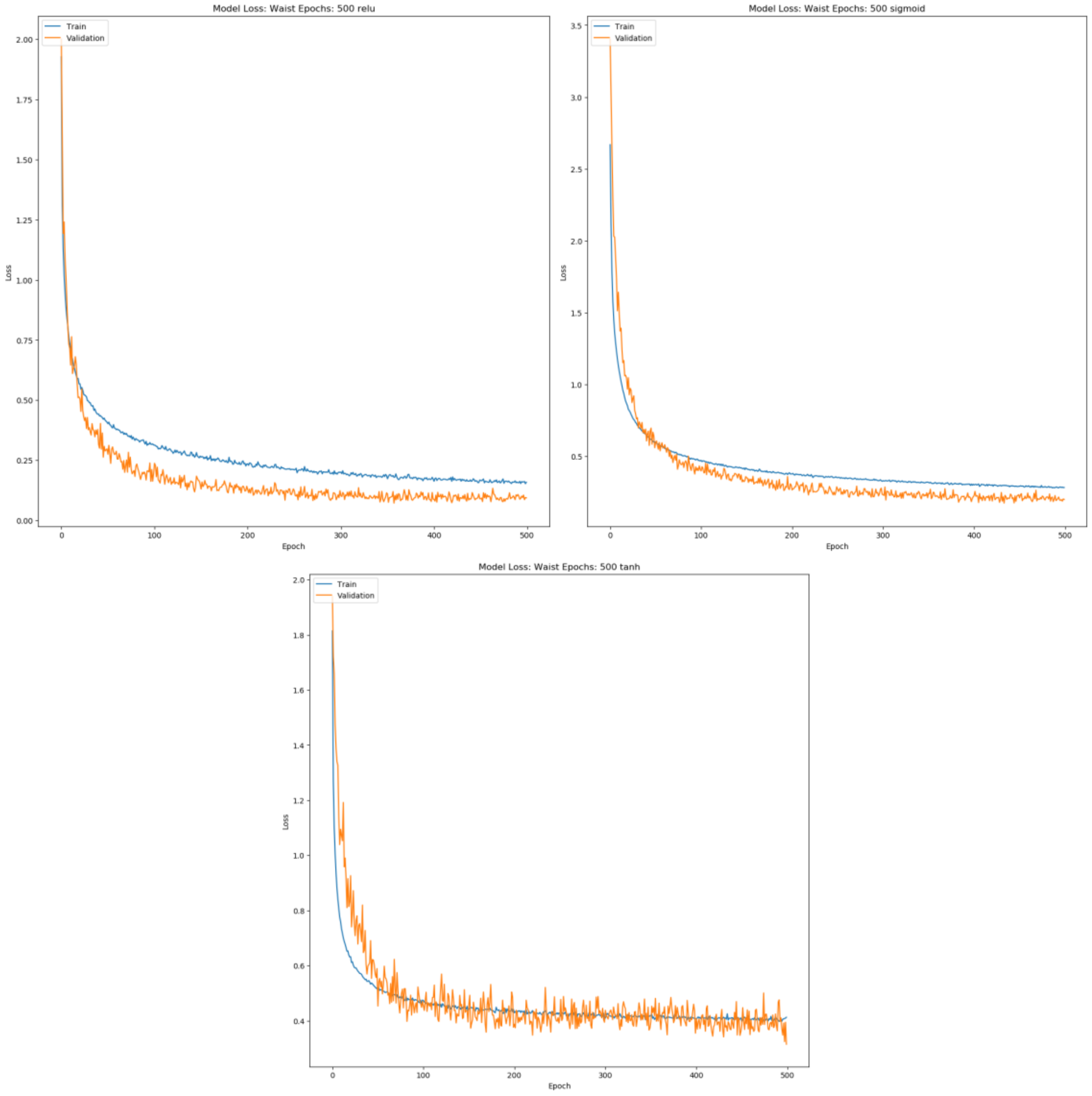


Figure 6-21: Training and Validation Loss at 500 Epochs function for Waist Data set.

From top-left: ReLu, Sigmoid, Tanh.

d) CNN Model Training at 1000 Epochs

Training and validation accuracy at 1,000 epochs emphasise the earlier comments of learning only increasing slightly at 500 epochs. We can see in Figure 6-22 that ReLu shows small improvements again, maintaining a training and validation accuracy at around 95%. Sigmoid does show a larger improvement here achieving a training and validation accuracy of 90%. This reflects in Sigmoid's testing accuracy where it achieved its highest results as shown in Table 6-3 suggesting that Sigmoid is more effective at higher epochs. Tanh does display an increase in training accuracy compared to 500 epochs. However, the validation accuracy on the wrist data set does begin to decrease. This suggests that Tanh begins to suffer from overfitting at this epoch. Testing accuracy is also lower, further highlighting this.

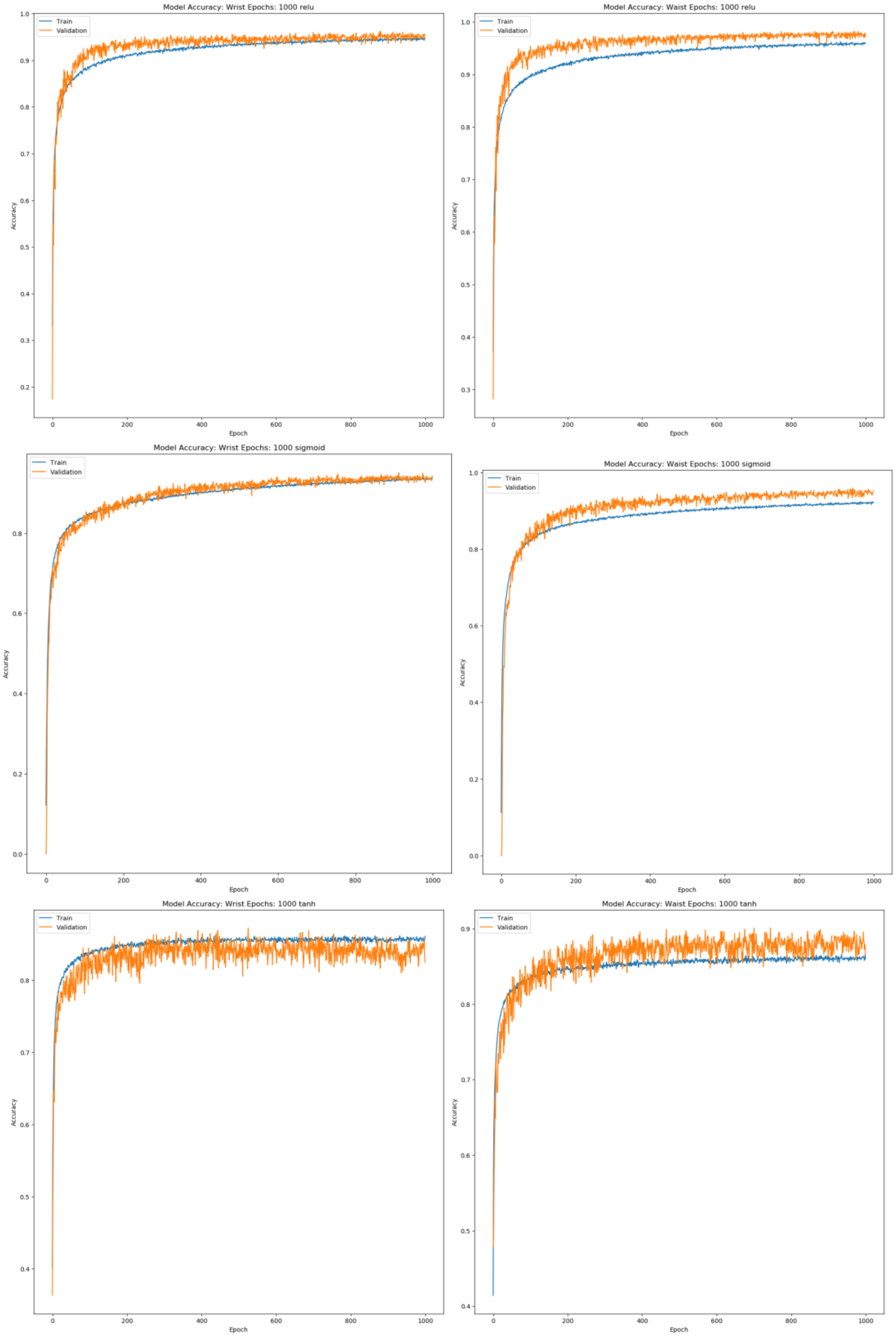


Figure 6-22: Training and Validation Accuracy at 1000 Epochs function for Waist Data set.

From Top: ReLu, Sigmoid, Tanh.

At 1,000 epochs, we can observe the wrist data set for ReLu begin to overfit as the validation loss begins to increase as shown in Figure 6-23. This can also be seen in Table 6-4 on the testing data set. This suggests that at a higher epoch, ReLu can no longer generalize the data as well as 200 epochs for example. However, validation loss on the waist data set does not begin to increase, therefore it may be only the activities performed on the wrist, which can involve more complex motions causing the model to begin overfitting rather than learning the data. Sigmoid also displays an increase in validation loss on the wrist data set, this can also be seen at 500 epochs as displayed in Appendix I. 1,000 epochs does appear to be where Sigmoid achieves its higher accuracies however, introducing additional data in the future may cause a loss in accuracy due to the overfitting observed here. Tanh begins to overfit again as shown in the validation loss on both wrist and waist data sets. This suggests that the 1,000 epochs is too many for this activation function as we do see a large increase on the testing loss as shown in Table 6-4. 500 epochs seem to be a sufficient amount for Tanh to provide a good classification of activities while minimizing loss.

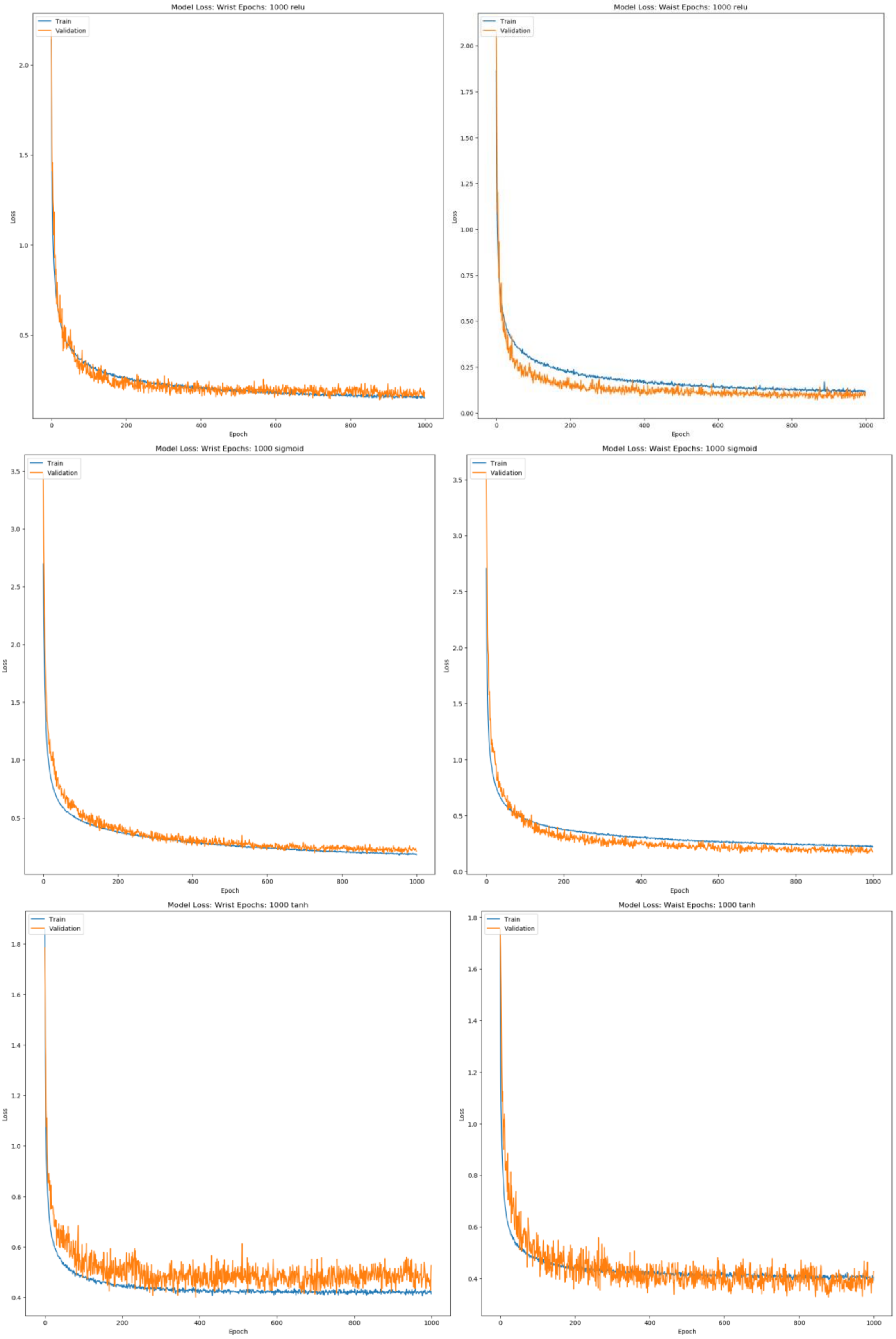


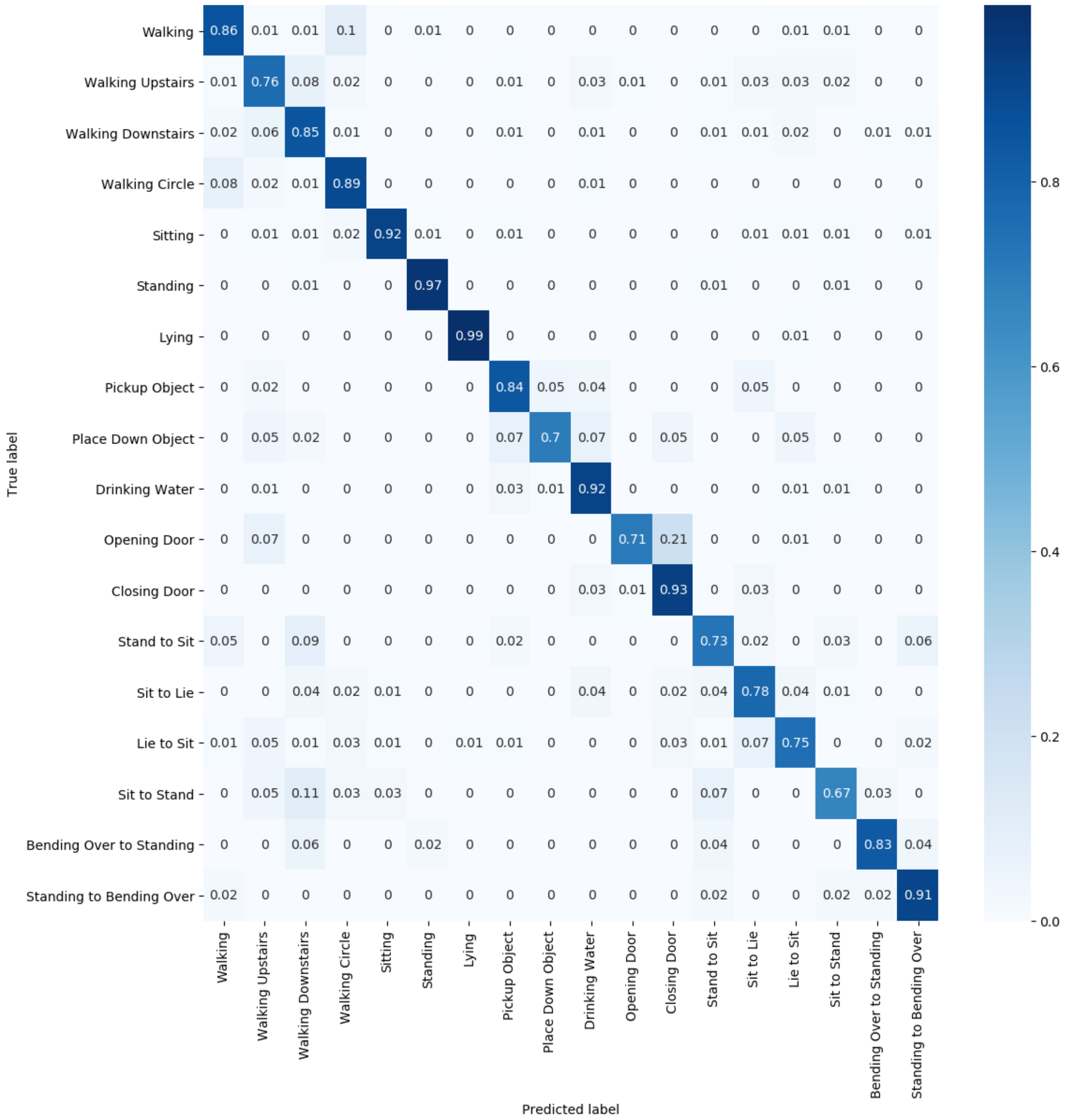
Figure 6-23: Training and Validation Loss at 1000 Epochs function for Waist Data set.

From Top: ReLu, Sigmoid, Tanh.

Figure 6-24 displays the confusion matrix for the wrist activity data using the ReLu activation function at 1,000 epochs. Compared to 200 epochs as shown in Figure 6-12 we can see that some activity types are classified better at 1,000 epochs and some begin to suffer. Walking activities do show an increase, with walking upstairs having the biggest accuracy gain where it is misclassified as similar activities such as walking downstairs and several transitional activities. Sedentary activities do show a decrease in accuracy at 1,000 epochs, with sitting falling to 92% accuracy compared to 99% at 200 epochs.

This potentially may be due to the overfitting we discussed earlier with the validation loss.

Remaining dynamic activities had improvements and accuracy losses. Picking up object had a small accuracy increase, however placing down object had a big loss in accuracy compared to 200 epochs with it mainly being misclassified as pickup object and drinking water. Opening door had a big loss as well with it mainly being misclassified as closing door which managed to make accuracy gains. Transitional activities were mostly harder to classify with sit-to-lie and standing-to-bend-over making the only gains. Other transitional activities suffered accuracy losses such as sit-to-stand which fell to 67% compared to 72% at 200 epochs. This was mainly misclassified as stand-to sit, most likely due to similar motions and standing positions and walking downstairs, possibly due to similar arm motions during the transition.



	Walking	Walking Upstairs	Walking Downstairs	Walking Circle	Sitting	Standing	Lying	Pickup Object	Place Down Object	Drinking Water	Opening Door	Closing Door	Stand to Sit	Sit to Lie	Lie to Sit	Sit to Stand	Bending Over to Standing	Standing to Bending Over
Walking	0.86	0.01	0.01	0.1	0	0.01	0	0	0	0	0	0	0	0	0.01	0.01	0	0
Walking Upstairs	0.01	0.76	0.08	0.02	0	0	0	0.01	0	0.03	0.01	0	0.01	0.03	0.03	0.02	0	0
Walking Downstairs	0.02	0.06	0.85	0.01	0	0	0	0.01	0	0.01	0	0	0.01	0.01	0.02	0	0.01	0.01
Walking Circle	0.08	0.02	0.01	0.89	0	0	0	0	0	0.01	0	0	0	0	0	0	0	0
Sitting	0	0.01	0.01	0.02	0.92	0.01	0	0.01	0	0	0	0	0	0.01	0.01	0.01	0	0.01
Standing	0	0	0.01	0	0	0.97	0	0	0	0	0	0	0.01	0	0	0.01	0	0
Lying	0	0	0	0	0	0	0.99	0	0	0	0	0	0	0	0.01	0	0	0
Pickup Object	0	0.02	0	0	0	0	0	0.84	0.05	0.04	0	0	0	0.05	0	0	0	0
Place Down Object	0	0.05	0.02	0	0	0	0	0.07	0.7	0.07	0	0.05	0	0	0.05	0	0	0
Drinking Water	0	0.01	0	0	0	0	0	0.03	0.01	0.92	0	0	0	0	0.01	0.01	0	0
Opening Door	0	0.07	0	0	0	0	0	0	0	0	0.71	0.21	0	0	0.01	0	0	0
Closing Door	0	0	0	0	0	0	0	0	0	0.03	0.01	0.93	0	0.03	0	0	0	0
Stand to Sit	0.05	0	0.09	0	0	0	0	0.02	0	0	0	0	0.73	0.02	0	0.03	0	0.06
Sit to Lie	0	0	0.04	0.02	0.01	0	0	0	0	0.04	0	0.02	0.04	0.78	0.04	0.01	0	0
Lie to Sit	0.01	0.05	0.01	0.03	0.01	0	0.01	0.01	0	0	0	0.03	0.01	0.07	0.75	0	0	0.02
Sit to Stand	0	0.05	0.11	0.03	0.03	0	0	0	0	0	0	0	0.07	0	0	0.67	0.03	0
Bending Over to Standing	0	0	0.06	0	0	0.02	0	0	0	0	0	0	0.04	0	0	0	0.83	0.04
Standing to Bending Over	0.02	0	0	0	0	0	0	0	0	0	0	0	0.02	0	0	0.02	0.02	0.91

Figure 6-24: Wrist Data set Confusion Matrix for ReLu at 1000 Epochs

Figure 6-25 shows the wrist data set confusion matrix for the model using Sigmoid at 1,000 epochs. We can observe that trends, mentioned previously, of accuracy improving for Sigmoid as epochs are increased. Compared to 200 epochs as shown in Appendix Jwith the exception of walking circle which makes a small accuracy loss, we see an increased accuracy for the remaining walking activities, with walking making the largest gains. The vast majority of misclassifications for walking activities seem to be other walking activities. Sitting and standing also make small gains, maintaining a high accuracy. The remaining dynamic activities display large accuracy increases for picking up object, placing down object and drinking water. Drinking water appears to be mostly misclassified as picking up object and placing down object, however it is also misclassified as sitting. The most likely cause for this will be due to the participant sitting down when they perform this activity.

Opening door does lose accuracy at 1,000 epochs with an increased misclassification as closing door and sit-to-lie. Transitional activities also made improvements with the exception of sit-to-lie and sit-to-stand which did suffer a decrease. Sit-to-stand appears to have been misclassified as the walking activities as well as stand-to-sit and sit-to-lie. For the transitional activities this is mostly likely due to the sitting element of all three activities being similar. For the walking activities, participants could have performed similar motions when standing up as they do when they are walking.

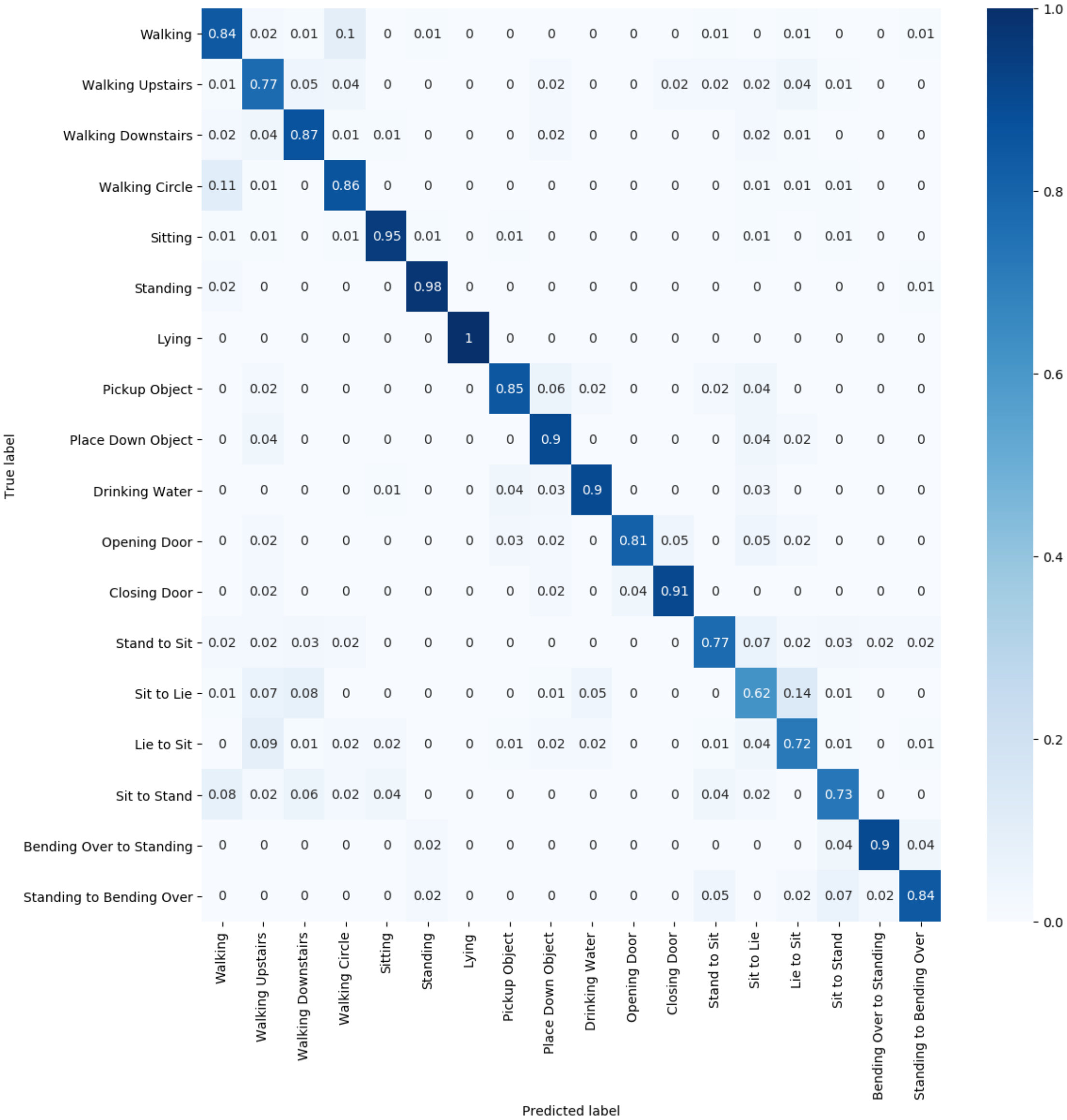


Figure 6-25: Wrist Data set Confusion Matrix for Sigmoid at 1000 Epochs

6.4.2. Evaluation of the Variations of the CNN Model at 200 Epochs

Additional tests of the CNN model were performed at 200 epochs, adjusting certain parameters in order to compare the performance of these variations against the results we discussed earlier. Firstly, we changed the final dense layer of the model to use the activation function used on the previous three layers rather than Softmax. This had a massively negative result for ReLu and Tanh which managed to achieve an extremely low accuracy as shown in Table 6-5. Sigmoid does manage to achieve a higher accuracy on both data sets compared to running at 200 epochs with Softmax on the final layer. We also adjusted the batch sizes, with the size changing to 32 and 128. ReLu only has a 1% accuracy difference on 32 batch sizes, and a 1% accuracy decrease on the wrist data set at 128. Sigmoid shows an improvement in both batch sizes over the 64 batch size we used for the main iteration of our model. Tanh shows a decrease in accuracy on the wrist data set for both batch sizes compared to the results shown in Table 6-3. There is a small increase in the waist set at batch size 32.

Table 6-5: Total Accuracy (%) of each model variant evaluation produced using CNN.

Accuracy %						
	ReLu		**Sigmoid**		**Tanh**	
Epoch: 200	**Wrist**	**Waist**	**Wrist**	**Waist**	**Wrist**	**Waist**
No Softmax	0.082319	0.082319	0.830725	0.845217	0.063188	0.117101
Batch Size 32	0.85971	0.864928	0.832464	0.851014	0.778551	0.824928
Batch Size 128	0.856232	0.857391	0.835942	0.844058	0.783188	0.811594

Loss for the model variations can be seen in Table 6-6.

ReLu and Tanh naturally have a higher loss when not using Softmax as the model cannot optimize itself in order to produce a high accuracy rate.

Sigmoid has a lower loss compared to the model with Softmax indicating that Sigmoid has had better optimization without. A reason for this, will be that Softmax can be considered an extension of Sigmoid for multi-classification cases. Therefore, Sigmoid still can maintain its ability to generalize the data compared to ReLu and Tanh.

ReLu shows increases and decrease to loss when changes to the batch size are applied. The wrist data set shows an increase in loss at 32 but a drop in loss for the waist data set. This shows a decrease in optimization of batch size 64 for the wrist and an optimization increase for the waist. 128 shows a decrease on the wrist data set and an increase for the waist. Sigmoid shows a decrease in loss for both batch sizes displaying better optimization over batch size 64 at 200 epochs. Tanh shows an increase in loss for batch size 128 and on the wrist data set at batch size 32. The model can optimize the waist data set better at batch size 32.

Table 6-6: Total Loss of each model variant evaluation produced using CNN.

Loss						
	ReLu		**Sigmoid**		**Tanh**	
Epoch: 200	**Wrist**	**Waist**	**Wrist**	**Waist**	**Wrist**	**Waist**
No Softmax	2.890372	2.890372	0.542211	0.496793	2.890372	2.890372
Batch Size 32	0.5035	0.427266	0.516572	0.463659	0.706893	0.568612
Batch Size 128	0.46861	0.487841	0.531991	0.496019	0.728564	0.615168

a) CNN Model Training without Softmax Applied

The histogram of kernel weights for the wrist data set without Softmax can be seen in Figure 6-26. On all four layers of the model, the weights stop being adjusted after around 20 epochs. As a result, the model has stopped learning and has begun underfitting. This is resulting in the model not being able to generalize the data set resulting in the poor accuracy we receive, and also not being able to classify the data as ReLu struggles with classification ability.

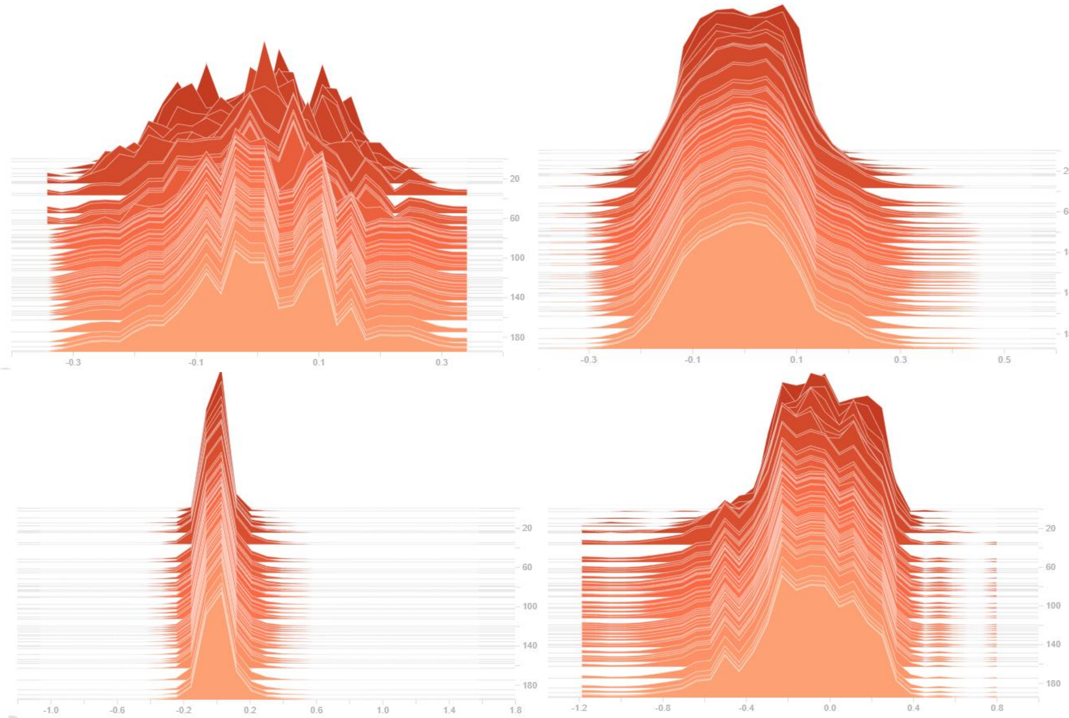


Figure 6-26: Wrist data set Histogram of kernel weights for the four layers of the network using ReLu at 200 epochs with no Softmax.

From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.

We can also see the high values of loss for Tanh in Figure 6-27. There is no curve as such, reducing the training and validation loss. For training, the loss plummets to about 2 and stays there before increasing again at about 110 epochs on the wrist and 150 on the waist. The validation set also has a plummet initially before constantly increasing and decreasing as time passes before it flat-lines. This emphasises the lack of ability for Tanh to generalize, resulting in the loss of accuracy. This could be due to Tanh having a focus on binary classification, meaning in our case, Softmax would be required in order for the model to perform multi-classification.

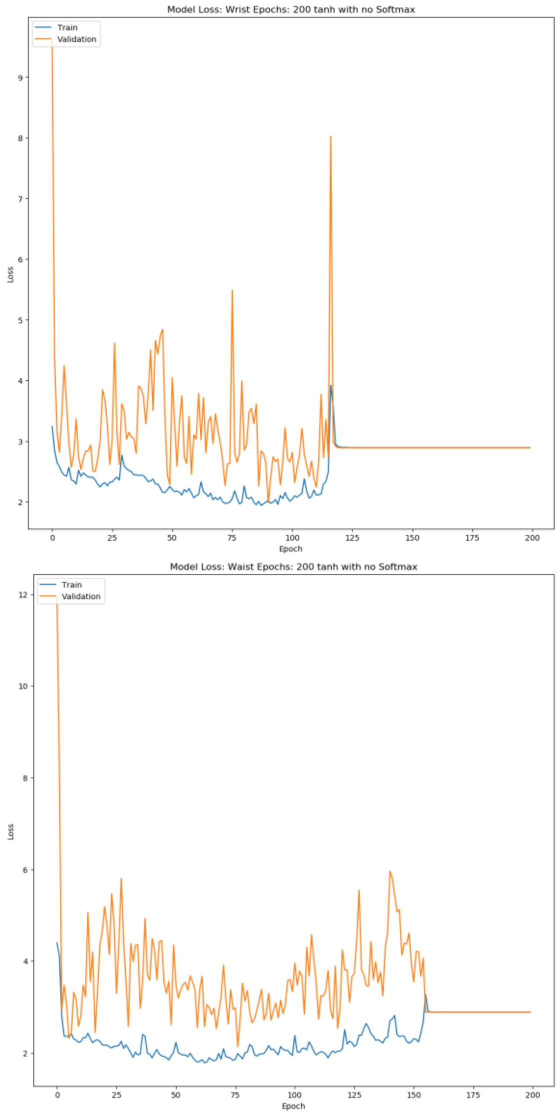


Figure 6-27: Wrist and waist loss for Tanh without Softmax at 200 Epochs

Sigmoid showed overall accuracy improvements without the use of Softmax. This is reflected in accuracy improvements for several activities on the wrist data set as shown in Figure 6-28. Walking activities showed a decrease in accuracy with the exception of walking. These tended to be misclassified as other walking activities. Sitting displayed a small improvement with the activity having been misclassified as several transitional activities. The remaining dynamic activities, with the exception of opening door had accuracy improvements over Sigmoid with Softmax. Closing door improved to a 91% accuracy and was mostly misclassified as opening door or drinking water. Drinking water had the largest increase, achieving 90% accuracy, with it mainly being misclassified as picking up and placing down objects. Transitional activities showed increases and decreases in accuracy over Sigmoid with Softmax. Bend-over-to-stand and stand-to-bend-over both improved. However, sit-to-stand showed a decrease from 82% to 69% with additional misclassifications occurring on stand-to-sit and walking activities.

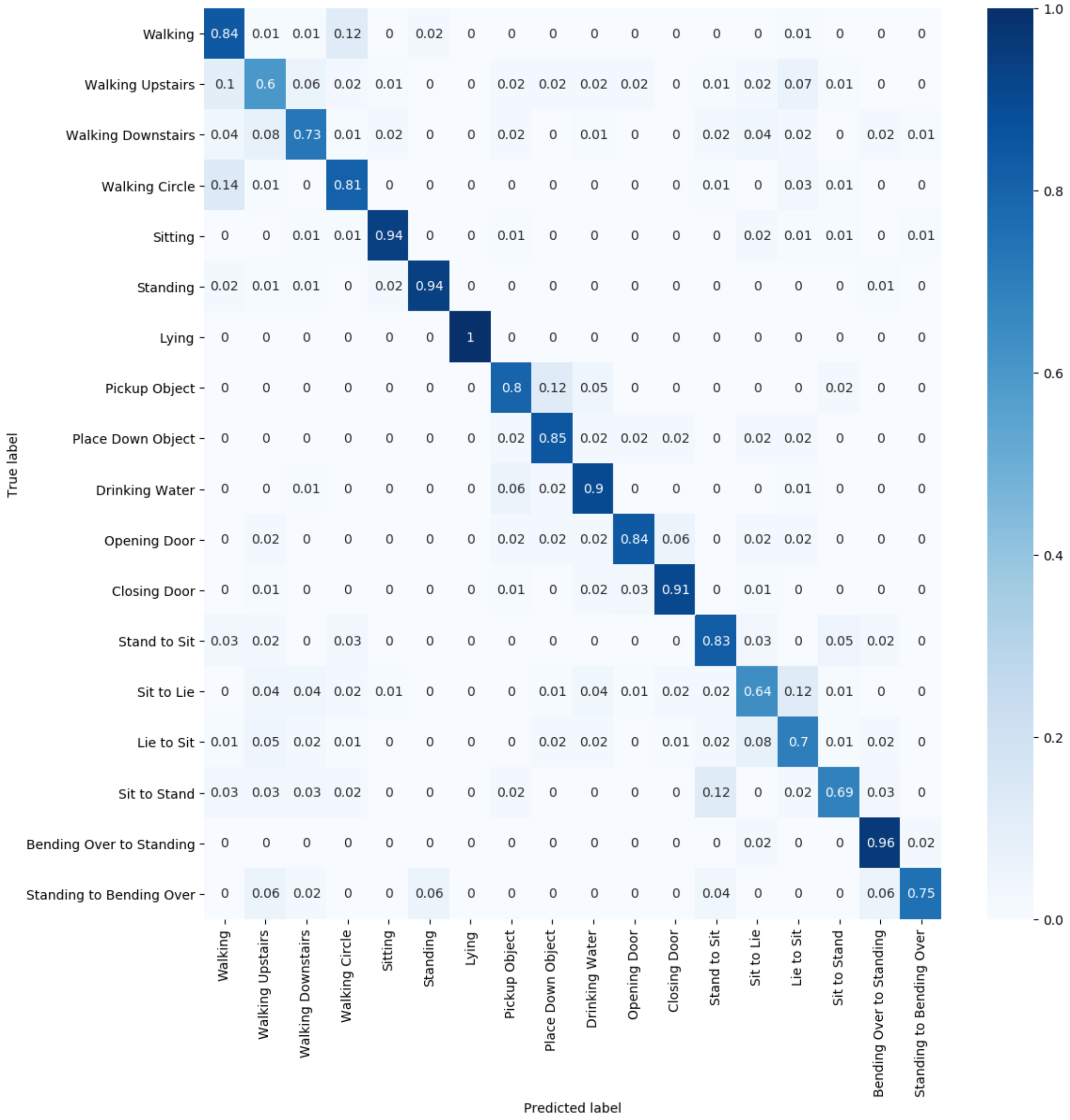
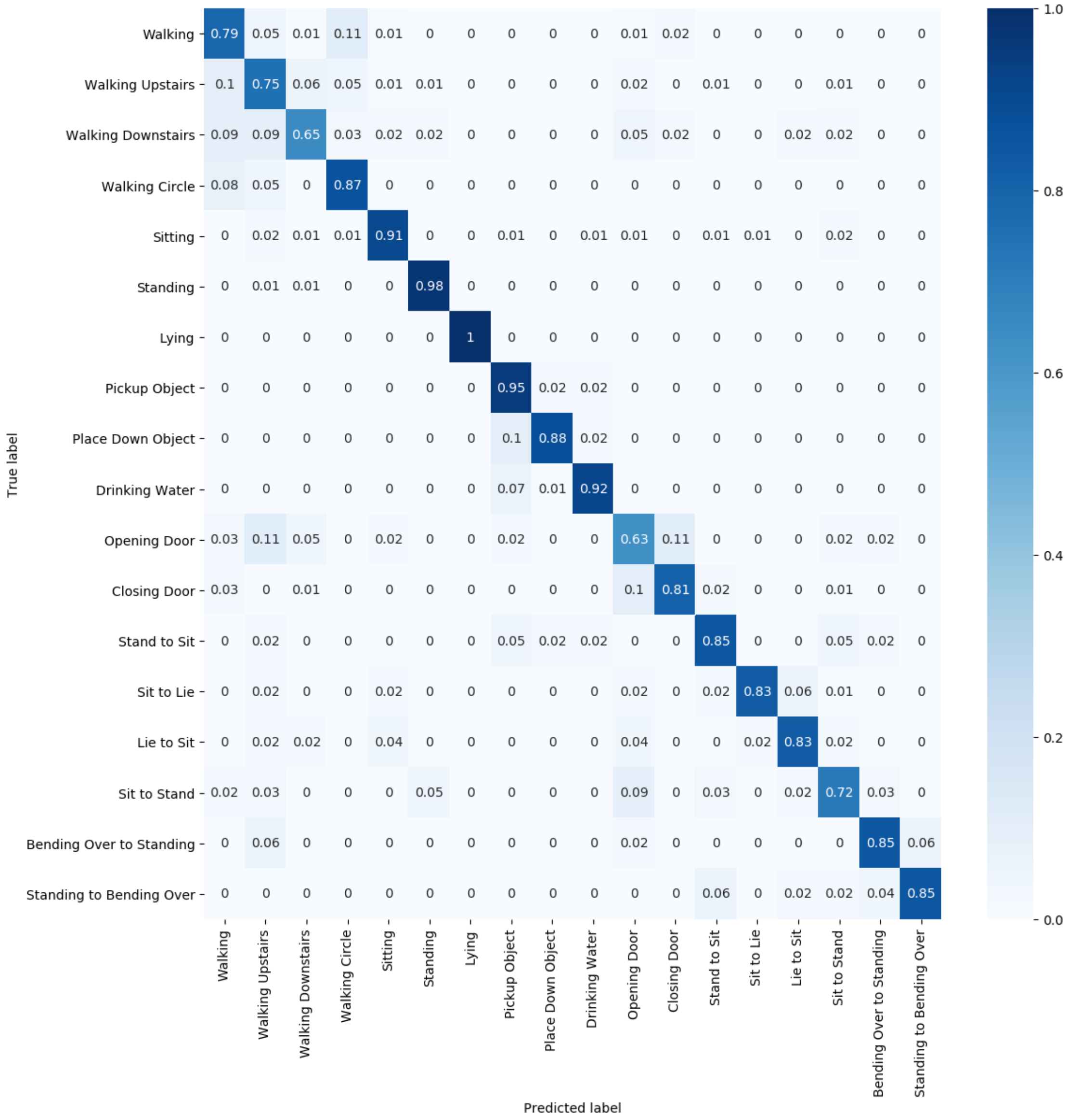


Figure 6-28: Wrist Data set Confusion Matrix for Sigmoid without Softmax at 200 Epochs

The waist data set also showed improvements in accuracy for most activities as shown in Figure 6-29. Walking and walking upstairs improved slightly, while walking downstairs and walking circle were harder to classify. Sedentary activities had some improvement, although lying remained at 100% accuracy. The remaining dynamic activities had some improvements with opening door, closing door, and place down object increasing accuracy slightly. Picking up object was harder to classify with misclassifications occurring as place down object. Many transitional activities were also harder to classify, sit-to-lie dropped to 64% accuracy compared to 86% on Sigmoid with Softmax. This was mainly misclassified as other transitional activities and some walking; sit-to-stand remained the same however, it was misclassified as different activities such as walking and walking upstairs.



	Walking	Walking Upstairs	Walking Downstairs	Walking Circle	Sitting	Standing	Lying	Pickup Object	Place Down Object	Drinking Water	Opening Door	Closing Door	Stand to Sit	Sit to Lie	Lie to Sit	Sit to Stand	Bending Over to Standing	Standing to Bending Over
Walking	0.79	0.05	0.01	0.11	0.01	0	0	0	0	0	0.01	0.02	0	0	0	0	0	0
Walking Upstairs	0.1	0.75	0.06	0.05	0.01	0.01	0	0	0	0	0.02	0	0.01	0	0	0.01	0	0
Walking Downstairs	0.09	0.09	0.65	0.03	0.02	0.02	0	0	0	0	0.05	0.02	0	0	0.02	0.02	0	0
Walking Circle	0.08	0.05	0	0.87	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sitting	0	0.02	0.01	0.01	0.91	0	0	0.01	0	0.01	0.01	0	0.01	0.01	0	0.02	0	0
Standing	0	0.01	0.01	0	0	0.98	0	0	0	0	0	0	0	0	0	0	0	0
Lying	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Pickup Object	0	0	0	0	0	0	0	0.95	0.02	0.02	0	0	0	0	0	0	0	0
Place Down Object	0	0	0	0	0	0	0	0.1	0.88	0.02	0	0	0	0	0	0	0	0
Drinking Water	0	0	0	0	0	0	0	0.07	0.01	0.92	0	0	0	0	0	0	0	0
Opening Door	0.03	0.11	0.05	0	0.02	0	0	0.02	0	0	0.63	0.11	0	0	0	0.02	0.02	0
Closing Door	0.03	0	0.01	0	0	0	0	0	0	0	0.1	0.81	0.02	0	0	0.01	0	0
Stand to Sit	0	0.02	0	0	0	0	0	0.05	0.02	0.02	0	0	0.85	0	0	0.05	0.02	0
Sit to Lie	0	0.02	0	0	0.02	0	0	0	0	0	0.02	0	0.02	0.83	0.06	0.01	0	0
Lie to Sit	0	0.02	0.02	0	0.04	0	0	0	0	0	0.04	0	0	0.02	0.83	0.02	0	0
Sit to Stand	0.02	0.03	0	0	0	0.05	0	0	0	0	0.09	0	0.03	0	0.02	0.72	0.03	0
Bending Over to Standing	0	0.06	0	0	0	0	0	0	0	0	0.02	0	0	0	0	0	0.85	0.06
Standing to Bending Over	0	0	0	0	0	0	0	0	0	0	0	0	0.06	0	0.02	0.02	0.04	0.85

Figure 6-29: Waist Data set Confusion Matrix for Sigmoid without Softmax at 200 Epochs

b) CNN Model Training with a 32 Batch Size

With a batch size of 32, ReLu training accuracy on the wrist data set reaches roughly 90% as we can see in Figure 6-30, this is roughly the same as with a batch size of 64. However, the validation accuracy is far more erratic, and appears to achieve as high a result, reducing the generalization ability. Sigmoid doesn't have a big change between a batch size of 64 and 32. Both validation and training accuracy reach a similar value. Tanh maintains a similar training accuracy to batch size of 64 however, its validation accuracy suffers and begins to decrease more noticeably, suggesting that Tanh struggles to generalize at this epoch is higher with a smaller batch size.

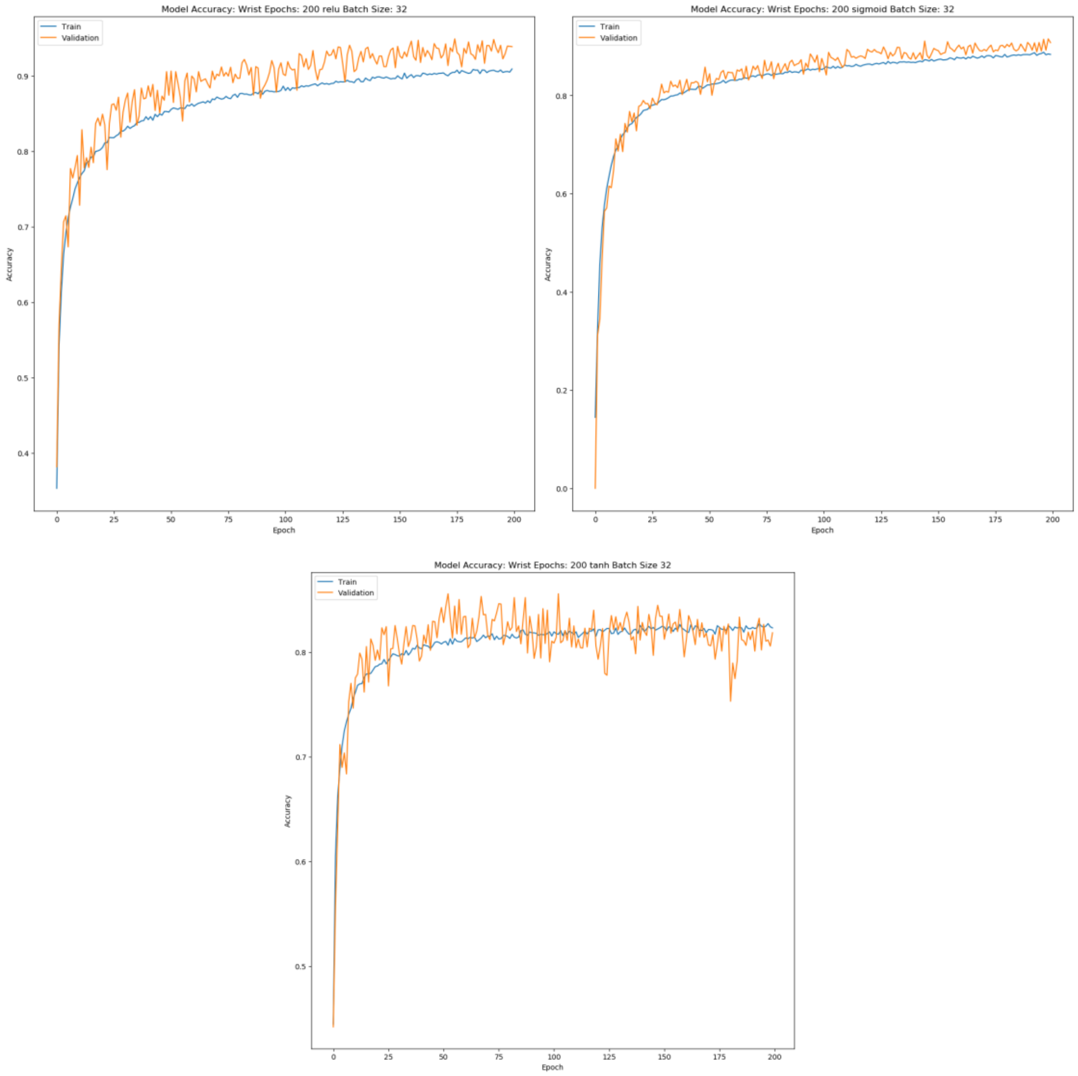


Figure 6-30: Wrist data set accuracy for activations functions at 200 epochs with a batch size of 32.

Loss for the wrist data set can be seen in Figure 6-31. ReLu shows that it is still learning and generalizing, the loss remains roughly the same as batch size 64. Sigmoid achieves a similar training loss compared to batch size 64, however the validation loss is lower indicating improved generalization. Tanh's validation loss also increases, matching with its drop in accuracy emphasising its potential overfitting, this appears to be more of an issue at 32 batch sizes.

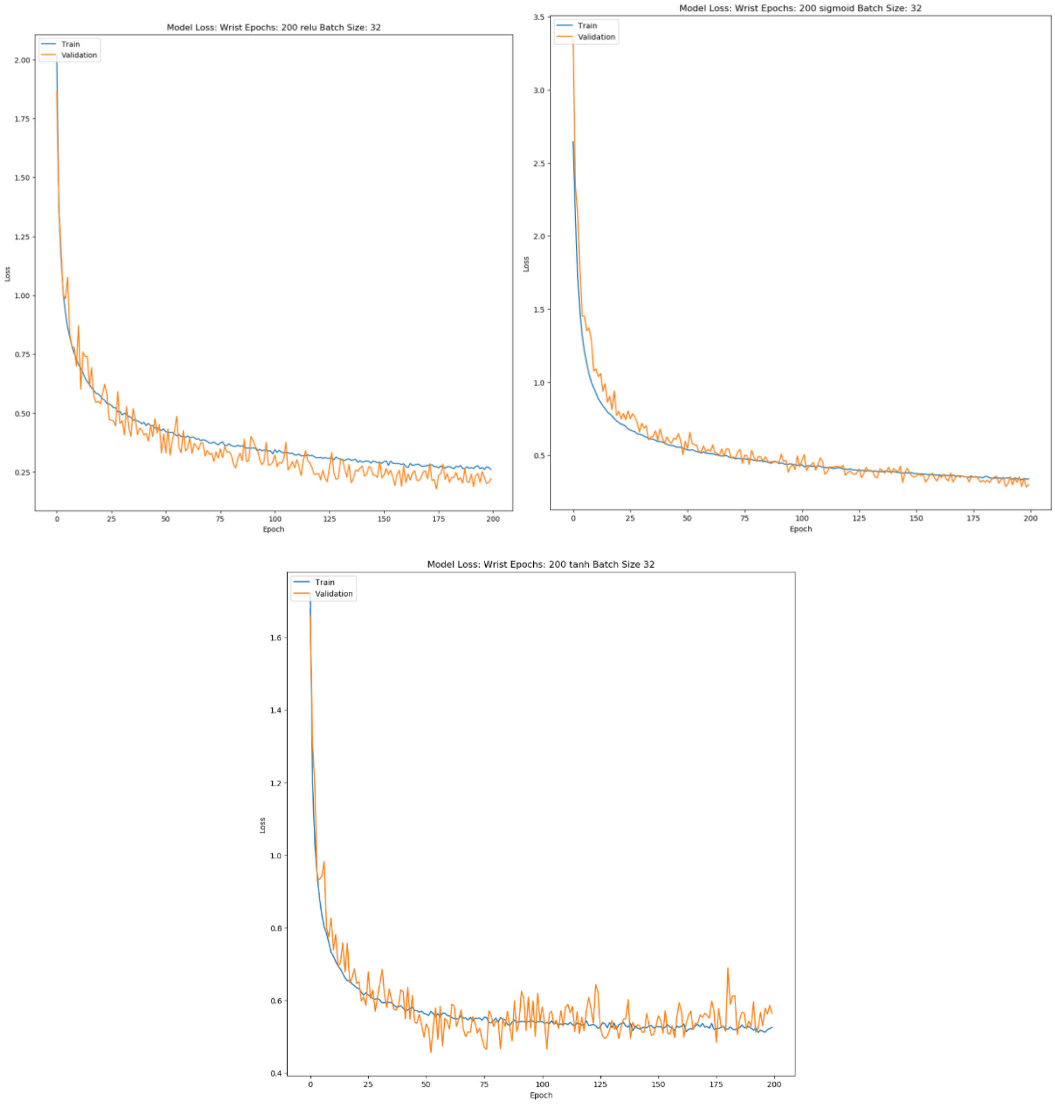


Figure 6-31: Wrist data set loss for activations functions at 200 epochs with a batch size of 32.

c) CNN Model Training with a 128 Batch Size

Figure 6-32 displays the accuracy of the waist data set using the activation functions with a batch size of 128. Compared to the batch size of 64, ReLu shows a slightly lower training accuracy and roughly the same validation accuracy. This indicates that the learning rate is slightly lower than batch size 64. Sigmoid maintains roughly the same training and validation accuracy. Tanh shows a lower training accuracy compared to batch size 64, and a less erratic validation accuracy which also remains fairly close to 90% and maintains good generalization.

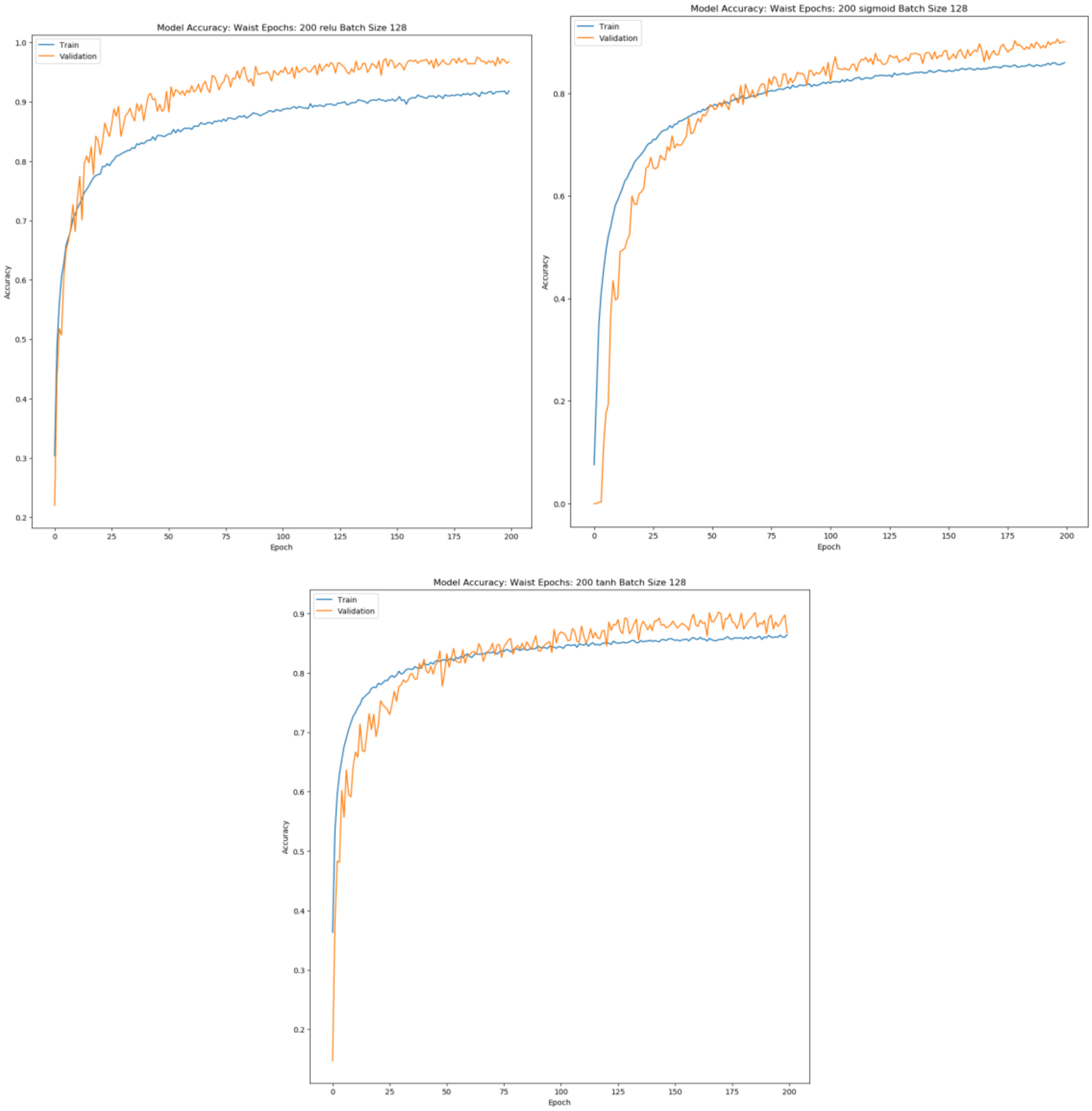


Figure 6-32: Waist data set accuracy for activations functions at 200 epochs with a batch size of 128.

Loss for batch size 128 can be seen in Figure 6-33. ReLu has a similar loss to that of 64 batch sizes. As generalization has remained the same, this indicates we haven't altered the batch size enough to see any noticeable change in the ability to learn or to generalize. Sigmoid also has a similar training and validation accuracy. Tanh doesn't show an increase in loss on the waist data compared to the wrist. However, compared to a batch size of 64, this remains roughly the same.

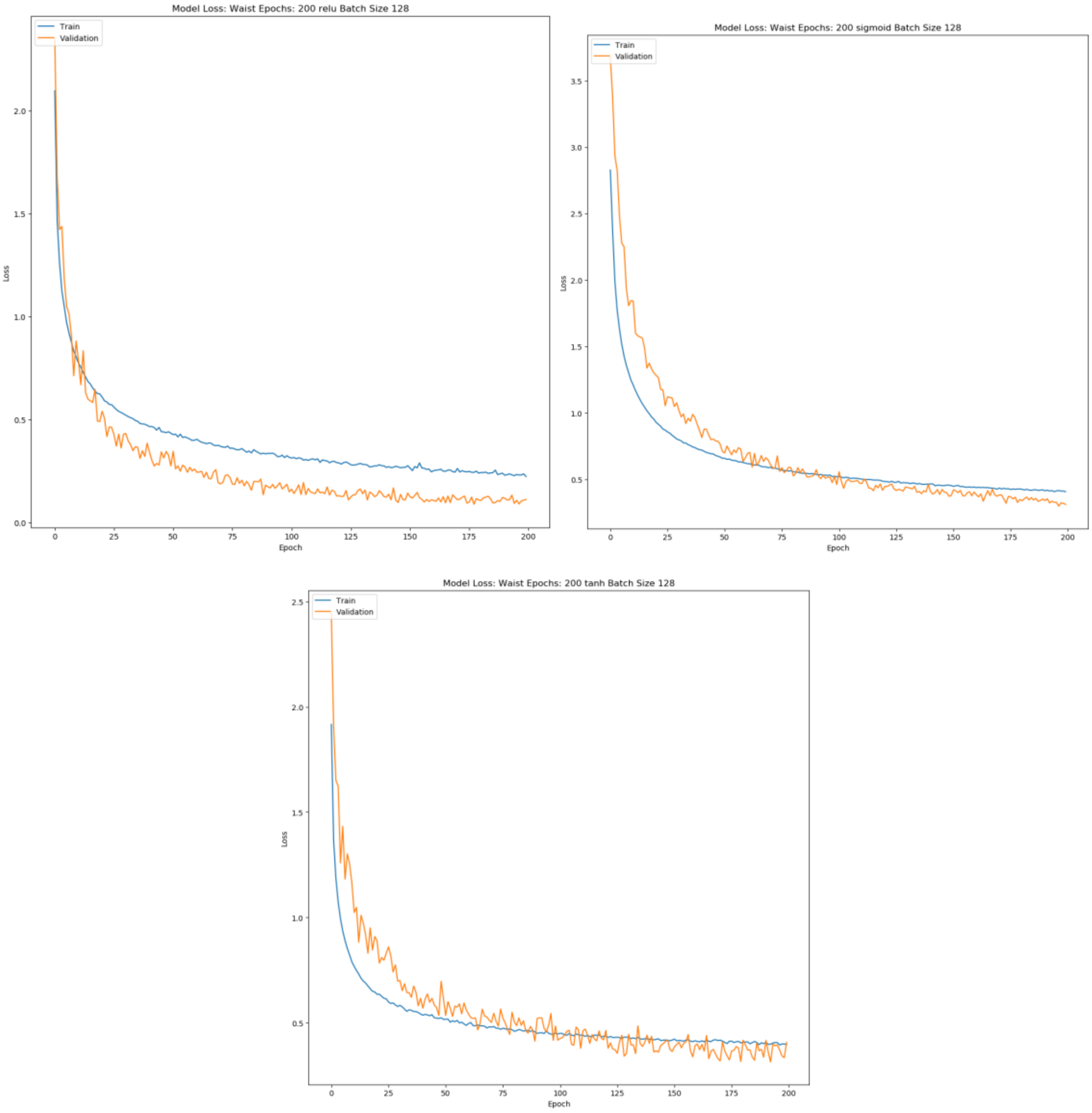


Figure 6-33: Waist data set loss for activations functions at 200 epochs with a batch size of 128.

6.4.3. Discussion

When choosing an appropriate model for PA classification, we must ensure that the accuracy is high and that it can be achieved in a time efficient manner. As we have seen with our model, training and validation accuracy has improved over time in some instances with overfitting beginning to occur for ReLu and Tanh activation functions as they approach 1,000 epochs. Sigmoid performs better at a high epoch, however this performance appears to be matched by ReLu at 200 epochs and with a lower loss. Tanh was more optimal at around 500 epochs, however its loss was still higher than what the other two activations function had, and its accuracy was also lower.

Sigmoid struggled to classify some transitional and dynamic activities at lower epochs. Our results show an improvement to sigmoid performance as epochs were increased, with accuracy rates increasing for each activity type. However, some dynamic and transitional activities still produced only average results, even at 1,000 epochs. The results also did not show a significant improvement over ReLu which performed better at lower epochs. The additional time required achieve a high accuracy when using Sigmoid would not be feasible in a clinical setting.

Tanh struggled with dynamic and transitional activities at lower epochs. Our results show the model overfitting when running with Tanh at 100 epochs. Overfitting was not present at 200 epochs, however the accuracy obtained was only average for dynamical and transitional activities compared to ReLu and Sigmoid. Accuracy with Tanh improved at 500 epochs; however, several transitional activities still displayed a low accuracy compared to the other activation functions. Our results show that at 1,000 epochs, the model begins to overfit again when using Tanh.

We can also see a decrease in activity accuracy for dynamic activities. Our model, when using Tanh, could not show overall higher accuracy rates and improvements to individual activity rates even at its peak performance when compared to ReLu and Sigmoid. This would not be suitable for a clinical setting when additional time is required to achieve the peak performance at 500 epochs compared to 200 epochs for ReLu and also as it cannot achieve a high accuracy for classification of activities when compared against ReLu and Sigmoid.

ReLu coped well with most activities at 200 epochs, with slight improvements as epochs increased. This did lead to the highest result for the waist data set at 1,000 epochs, however with 3% accuracy difference the additional time to compute may not be worth it. Instead, we could increase our accuracy by reducing our limitations. Participant numbers were difficult to increase which greatly limited the amount of data we could collect. By improving our data set, we could improve the accuracy we can achieve at 200 epochs with ReLu.

ReLu did show good performance on each activity type, particularly sedentary activities. It also displayed good classification ability on transitional activities which are typically harder to classify due to the more complex and subtle motions that occur as a person performs them. We can also observe the accuracy of classified activities begin to decrease for some types as the epochs were increased. Our results show that the models learning slows after around 200 epochs, and the loss begins to increase showing that the model begins to overfit as it draws closer to 1,000 epochs.

We have also shown an improvement over the initial study described in chapter 4.

By adding additional features and using data from an additional sensor from the Shimmer device, we managed to improve on the accuracy we obtained in that chapter. This was also achieved with fewer epochs in some tests. The accuracy levels achieved show the value and performance of low-cost devices such as Shimmer in activity recognition studies. They have the capacity to be less intrusive for further long-term studies into AR.

6.5. Summary

Three participants participated in our study to perform 18 activities that are typically performed around the home, in order to provide contextually meaningful data.

These activities were performed in a supervised lab environment with two Shimmer3 IMU devices being attached to the wrist and waist of participants. Data was extracted from the Accelerometer and gyroscope sensors with suitable time and frequency domain features extracted from the data.

Extracted features for the wrist and waist data sets were used as the inputs for our CNN model described in chapter 5.

The model was executed at four different epoch variations (100, 200, 500 and 1,000 epochs) using three activation functions; ReLu, Sigmoid and Tanh.

These activation functions display varying performance over epochs, with Tanh struggling to classify as well as the other two activation functions, ReLu performing better at lower epochs while Sigmoid perfumed better at higher epochs.

We believe ReLu is the optimal function for our model running at around 200 epochs as it provides a high accuracy in a short computation time, despite having a small data set – making it high-suited to clinician settings.

Chapter VII: Conclusion

This thesis has focused on physical activity recognition and monitoring in the elderly using low-cost wearable devices. This is extremely important from a healthcare perspective as around 850,000 people currently live in the UK with some form of Dementia. A number of studies have been aimed at studying activity recognition in order to improve the quality of life of the elderly. The main contribution of this thesis, is the methodology for PAR using low-cost devices for the elderly or people who have begun to suffer with memory loss or have been diagnosed with Mild Cognitive Impairment (MCI).

Firstly, we performed a systematic literature review of IoT technologies used in PAR in chapter 2. This covered the sensing layer where we discussed studies that incorporated a variety of sensors to study elderly people and sufferers of dementia, depression and other chronic illnesses. We discussed the processing and application layers for IoT networks providing a summary of techniques and applications used in deploying wearable technologies into lab and home-based environments. Processing approaches, features and classification methods are described and compared for their performance, accuracy and flexibility when deployed for PAR. Finally, we discuss how these approaches can be applied to real world environments, and provide examples of studies where attempts have been made to produce such applications.

Chapter 3 describes our framework for investigating activity recognition in the elderly. We break down the types of activities that can potentially be performed while providing a discussion of the limitations and mobility issues that an elderly person can suffer from in order to make necessary considerations for our study. We discuss the advantages of studies in home-based environments as they can provide larger data sets while maintaining the quality of life for the elderly. We discuss the types of wearable sensors and how their deployment across the different areas of the body can provide a variety of data such as microphones, smartphones, ECG and accelerometers. Finally, we discuss potential classifiers that can be used in an activity recognition model as well as discussing features and processing techniques.

Six participants under the age of 50, with no signs of MCI or other forms of memory loss and mental illness performed 15 activities of daily living in a lab environment using two Shimmer3 IMU placed on the wrist and waist and a smartphone for data collection in chapter 4. Activities were classified as dynamic, sedentary and transitional and the raw accelerometer signals gathered from these activities was discussed. The raw activity data was also subject to data pre-processing techniques and feature extraction methods such as signal magnitude vector as well as sliding window. Chapter 5 discusses the classification techniques, ANN, SVM and DT that were used to classify the features extracted from the raw data. Each classifier was discussed using statistics generated from the classified data in order to measure classifier performance. Waist data typically had average performance while ANN and SVM struggled to classify wrist data. This was followed up with a discussion of our convolution neural network that we developed using Tensorflow and Keras. This was a four-layer network using 1-dimensional convolution layers and dense layers.

We recruited 3 participants through Join Dementia Research to perform 18 activities in a lab environment for our case study in chapter 6. The service works closely with healthcare professionals and social care workers to encourage people to participate in Dementia related research studies. Each participant was over the age of 60 and had memory loss with no diagnosis or had been diagnosed with MCI. Those with additional mental illnesses or who could not provide consent due to memory problems were excluded from the study. Data was collected using 2 Shimmer3 IMU attached to the wrist and waist. Accelerometer and gyroscope data was extracted for processing and feature extraction. The CNN model we discussed in chapter 5 was used to classify the activities using ReLu, Sigmoid and Tanh activation functions. These were run at multiple epochs in order to gauge the models performance. The model was evaluated and discussed, highlighting its learning and generalization capability.

7.1. Limitations

The PAR methods proposed have met the research aims and objectives for this research, however several limitations existed that would need addressing and investigating in future work.

Firstly, our data sets were limited to 6 healthy participants and 3 participants with memory loss. The lack of participants has had an effect on performance as more data is necessary to classify to a good standard. Callouts for healthy participants were requested however, limited response came in. When recruiting participants through Join Dementia, our searches revealed a potential 50 participants that matched our search criteria. Emails were sent to all participants, and a select few that requested to be telephoned for information were contacted. However, many of the profiles contacted through Join Dementia had been inactive for some time. A number of people responded to our request, informing us that symptoms had progressed and they were unable to participate, or that the distance needed to travel to the lab space was too far. The lab space was another limitation that we were bound to by ethical approval. In the future, expanding the ethical approval to include home-environment testing could expand our participant capacity as well as the potential activities that could be performed. Furthermore, this could expand the duration over which participants were tested, creating a long-term PAR study.

7.2. Future Endeavours

IoT technologies have improved the prospects of PAR providing huge benefits to the healthcare industry and for the elderly allowing them to improve their quality of life. Many challenges remain in order to improve IoT technologies further to expand the potential of life-logging data, home environment setups, privacy concerns and reducing potential irritation of sensor setups for elderly people. Further research is required to address the challenges PAR IoT studies face to ensure the technology is suitable for IoT environments.

Cost effective solutions to IoT technologies have made advances over the last decade. However, this is still limited in regards to long-term healthcare monitoring, especially for home environments. Wearing a sensor, such as the Shimmer devices we used in our study, for long periods could become uncomfortable, especially for the elderly. For someone suffering with dementia, they may be inclined to remove the device, which would disrupt data collection. Mobile phones do offer some solution to this, however they can easily be misplaced and forgotten or placed in a bag, effecting the raw data collected. Battery life for these devices provides uncertainty for data collection, especially in natural environments. Charging for these devices may not always be available, therefore they require a sufficient charge overnight or before any activity is performed. There is also a trade-off in performance when using low cost sensing technologies, which can impact the overall quality of data. Further work in expanding the capabilities of lost-cost devices in natural environments is key for the expansion of IoT technologies in the healthcare sector.

In uncontrolled environments, there is a variability in how subjects handle activities as an individual's performance and behaviours differ from others.

Physical characteristics such as age and weight play a part in this, however major uncertainties occur from complex PA, such as playing sports.

Standard mathematical models will struggle to cope with these varying time-based sequences therefore further investigation into optimizing frameworks is necessary for improving the stability of uncontrolled environment studies.

Memory capacity is an issue that effects the possibility of life-logging studies. IoT technologies allow for life-logging studies to be performed, however the vast quantity of data gathered in these studies provides the challenge of storage.

Many solutions have taken to using cloud technology; however, this comes with a price tag as well as privacy concerns. GPU technologies are also deployed in order to increase layer counts in models and to develop deep learning approaches in order to achieve high accuracy rates in the classification results. However, this comes at the cost of requiring specialist hardware which is unlikely to be readily available in a clinical environment. PAR studies have to address these challenges to ensure that their data storage processes meet the security concerns of participants, while achieving the required memory capacity.

References

[1] Alzheimer's Society, "About dementia," 2014. https://www.alzheimers.org.uk (accessed Aug. 10, 2016).

[2] S. Finkel and J. C. e Silva, "Behavioral and psychological signs and symptoms of dementia: a consensus statement on current knowledge and implications for research and treatment," *Int.* …, vol. 12, no. February, pp. 1060–1061, 1997, [Online]. Available: http://journals.cambridge.org/abstract_S1041610297003943.

[3] M. Yefimova and D. L. Woods, "Using Sensor Technology to Monitor Disruptive Behavior of Persons With Dementia," *AAAI Fall Symp. Ser. 2012 AAAI Fall Symp. Ser.*, pp. 51–54, 2012, [Online]. Available: http://www.aaai.org/ocs/index.php/FSS/FSS12/paper/view/5617/5791.

[4] J. Coppola, M. Kowtko, C. Yamagata, and S. Joyce, "Applying Mobile Application Development to Help Dementia and Alzheimer Patients," *Csis.Pace.Edu*, pp. 1–7, 2013, doi: 10.1109/LISAT.2013.6578252.

[5] S. Michie, "The Behaviour Change Wheel : a new method for characterising and designing behaviour change interventions Why focus on behaviour ?," vol. 42, no. April, 2015.

[6] M. Ziefle and R. Carsten, "Acceptance of Pervasive Healthcare Systems: A comparison of different implementation concepts," *Comput. Technol. Healthc. (*, pp. 1–6, 2010, doi: 10.4108/ICST.PERVASIVEHEALTH2010.8915.

[7] J. Healey, "Future possibilities in electronic monitoring of physical activity.," *Res. Q. Exerc. Sport*, vol. 71, no. September, pp. S137–S145, 2000, doi: 10.1080/02701367.2000.11082797.

[8] J. Qi, L. Chen, W. Leister, and S. Yang, "Towards knowledge driven decision support for personalized home-based self-management of chronic diseases," *Proc. - 2015 IEEE 12th Int. Conf. Ubiquitous Intell. Comput. 2015 IEEE 12th Int. Conf. Adv. Trust. Comput. 2015 IEEE 15th Int. Conf. Scalable Comput. Commun. 20*, pp. 1724–1729, 2016, doi: 10.1109/UIC-ATC-ScalCom-CBDCom-IoP.2015.313.

[9] J. Qi, P. Yang, M. B. Hanneghan, and S. Tang, "Multiple Density Maps Information Fusion for Effectively Assessing Intensity Pattern of Lifelogging Physical Activity," *Neurocomputing*, no. 286545, 2016, doi: 10.1016/j.neucom.2016.06.073.

[10] Q. Liu, W. Zhang, H. Li, and K. N. Ngan, "Hybrid human detection and recognition in surveillance," *Neurocomputing*, vol. 194, pp. 10–23, 2016, doi: 10.1016/j.neucom.2016.02.011.

[11] A. Jalal, M. Uddin, and T. S. Kim, "Depth video-based human activity recognition system using translation and scaling invariant features for life logging at smart home," *IEEE Trans. Consum. Electron.*, vol. 58, no. 3, pp. 863–871, 2012, doi: 10.1109/TCE.2012.6311329.

[12] P. Moore, F. Xhafa, L. Barolli, and A. Thomas, "Monitoring and detection of agitation in dementia: Towards real-time and big-data solutions," *Proc. - 2013 8th Int. Conf. P2P, Parallel, Grid, Cloud Internet Comput. 3PGCIC 2013*, pp. 128–135, 2013, doi: 10.1109/3PGCIC.2013.26.

[13] J. Pärkkä, M. Ermes, P. Korpipää, J. Mäntyjärvi, J. Peltola, and I. Korhonen, "Activity classification using realistic data from wearable sensors.," *IEEE Trans. Inf. Technol. Biomed.*, vol. 10, no. 1, pp. 119–128, 2006, doi: 10.1109/TITB.2005.856863.

[14] M. Ermes, J. Pärkkä, J. Mäntyjärvi, and I. Korhonen, “Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions,” *Saudi Med. J.*, vol. 32, no. 6, pp. 640–641, 2011, doi: 10.1109/TITB.2007.899496.

[15] S. I. Soraya *et al.*, “IoT/M2M wearable-based activity-calorie monitoring and analysis for elders,” *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, pp. 2390–2393, 2017, doi: 10.1109/EMBC.2017.8037337.

[16] T. Perumal and Y. L. Chui, “IoT Based Activity Recognition among Smart Home Resididents,” no. Gcce, pp. 1–2, 2017.

[17] H. Li, C. Y. Wan, R. C. Shah, A. P. Sample, and S. N. Patel, “IDAct: Towards unobtrusive recognition of user presence and daily activities,” *2019 IEEE Int. Conf. RFID, RFID 2019*, pp. 1–8, 2019, doi: 10.1109/RFID.2019.8719103.

[18] J. Qi, P. Yang, M. Hanneghan, D. Fan, Z. Deng, and F. Dong, “Ellipse fitting model for improving the effectiveness of life-logging physical activity measures in an Internet of Things environment,” pp. 107–113, 2016, doi: 10.1049/iet-net.2015.0109.

[19] P. Yang, M. Hanneghan, J. Qi, D. Fan, Z. Deng, and F. Dong, “Improving the Validity of Lifelogging Physical Activity Measures in an Internet of Things Environment,” *IET Networks*, vol. 5, no. 5, pp. 107–113, 2016, doi: 10.1049/iet-net.2015.0109.

[20] J. Qi, P. Yang, D. Fan, and Z. Deng, “A Survey of Physical Activity Monitoring and Assessment Using Internet of Things Technology,” *Comput. Inf. Technol. Ubiquitous Comput. Commun. Dependable, Auton. Secur. Comput. Pervasive Intell. Comput. (CIT/IUCC/DASC/PICOM), 2015 IEEE Int. Conf.*, pp. 2353–2358, 2015, doi: 10.1109/CIT/IUCC/DASC/PICOM.2015.348.

[21] B. Aguiar, T. Rocha, and J. Silva, “Accelerometer-Based Fall Detection for Smartphones,” 2014.

[22] T. Toutountzi and S. Phan, “A Framework for the Assessment of Wandering Behavior.”

[23] E. Batista, F. Borras, F. Casino, and A. Solanas, “A Study on the Detection of Wandering Patterns in Human Trajectories,” pp. 1–6.

[24] K. Kim, M. M. Hassan, S. Na, and E. Huh, “Dementia Wandering Detection and Activity Recognition Algorithm Using Tri-axial Accelerometer Sensors,” 2009.

[25] A. Solanas, E. Batista, F. Borras, and C. Patsakis, “Wandering Analysis with Mobile Phones On the Relation Between Randomness and Wandering.”

[26] F. Sposaro, J. Danielson, and G. Tyson, “iWander : An Android Application for Dementia Patients,” pp. 3875–3878, 2010.

[27] A. K. Bourke, J. V. O. Brien, and G. M. Lyons, “Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm,” vol. 26, pp. 194–199, 2007, doi: 10.1016/j.gaitpost.2006.09.012.

[28] Y. Zigel, D. Litvak, and I. Gannot, “A Method for Automatic Fall Detection of Elderly People Using Floor Vibrations and Sound — Proof of Concept on Human Mimicking Doll Falls,” vol. 56, no. 12, pp. 2858–2867, 2009.

[29] G. R. Yavuz *et al.*, “A Smartphone Based Fall Detector with Online Location Support.”

[30] J. Wang, Z. Zhang, B. Li, S. Lee, and R. S. Sherratt, “An Enhanced Fall Detection

System for Elderly Person Monitoring using Consumer Home Networks," pp. 23–29, 2014.

[31] F. G. Miskelly, "Assistive technology in elderly care.," *Age Ageing*, vol. 30, no. 6, pp. 455–458, 2001, doi: 10.1093/ageing/30.6.455.

[32] W. Azizul, S. Imai, and T. Hasegawa, "Fall detection using smart phone and ubiquitous pulse wave sensor," no. microS D, pp. 3–6.

[33] D. Miranda, M. Calderón, and J. Favela, "Anxiety detection using wearable monitoring," pp. 34–41, 2014.

[34] S. Ishimaru, K. Kunze, and K. Kise, "In the Blink of an Eye – Combining Head Motion and Eye Blink Frequency for Activity Recognition with Google Glass," *Ah*, pp. 15:1-15:4, 2014, doi: 10.1145/2582051.2582066.

[35] A. Bankole *et al.*, "Continuous, non-invasive assessment of agitation in dementia using inertial body sensors," *Proc. 2nd Conf. Wirel. Heal. - WH '11*, p. 1, 2011, doi: 10.1145/2077546.2077548.

[36] H. Lu *et al.*, "StressSense: Detecting Stress in Unconstrained Acoustic Environments using Smartphones," *Proc. 2012 ACM Conf. Ubiquitous Comput. - UbiComp '12*, p. 351, 2012, doi: 10.1145/2370216.2370270.

[37] J. H. L. Hansen and S. Bou-Ghazale, "Getting started with SUSAS: A speech under simulated and actual stress database," *Proc EuroSpeech*, 1997, [Online]. Available: http://www.ldc.upenn.edu/Catalog/docs/LDC99S78/susas_rev1b4.ps%5Cnpapers2://publication/uuid/88496064-9CE2-479D-AB86-4C8C697BF578.

[38] R. Fernandez and R. W. Picard, "Modeling drivers' speech under stress," *Speech Commun.*, vol. 40, no. 1–2, pp. 145–159, 2003, doi: 10.1016/S0167-6393(02)00080-8.

[39] S. A. Patil and J. H. Hansen, "Detection of speech under physical stress: model development, sensor selection, and feature fusion," *Interspeech*. 2008.

[40] V. Foo *et al.*, "Automated Recognition of Complex Agitation Behavior of Dementia Patients Using Video Camera," pp. 68–73, 2007.

[41] A. Gruenerbl *et al.*, "Using smart phone mobility traces for the diagnosis of depressive and manic episodes in bipolar patients," *Proc. 5th Augment. Hum. Int. Conf.*, p. 38, 2014, doi: 10.1145/2582051.2582089.

[42] L. Canzian and M. Musolesi, "Trajectories of depression," *Proc. 2015 ACM Int. Jt. Conf. Pervasive Ubiquitous Comput. - UbiComp '15*, pp. 1293–1304, 2015, doi: 10.1145/2750858.2805845.

[43] M. N. Burns *et al.*, "Harnessing context sensing to develop a mobile intervention for depression," *J. Med. Internet Res.*, vol. 13, no. 3, pp. 1–17, 2011, doi: 10.2196/jmir.1838.

[44] R. F. Dickerson, T. Hnat, E. Hoque, and J. a Stankovic, "Demonstration of sleep monitoring and caregiver displays for depression monitoring," *Proc. 2nd Conf. Wirel. Heal. - WH '11*, p. 1, 2011, doi: 10.1145/2077546.2077571.

[45] J. R. Williamson *et al.*, "Detecting Depression using Vocal, Facial and Semantic Communication Cues," *Proc. 6th Int. Work. Audio/Visual Emot. Chall. - AVEC '16*, pp. 11–18, 2016, doi: 10.1145/2988257.2988263.

[46] P. Loventoft, L. Norregaard, and E. Frokjær, "Designing daybuilder: an experimental app to support people with depression," *Proc. 12th …*, pp. 1–4, 2012,

doi: 10.1145/2348144.2348146.

[47] Z. Liu *et al.*, “Detection of depression in speech,” *2015 Int. Conf. Affect. Comput. Intell. Interact. ACII 2015*, pp. 743–747, 2015, doi: 10.1109/ACII.2015.7344652.

[48] A. Doryab, J. K. Min, J. Wiese, J. Zimmerman, and J. I. Hong, “Detection of behavior change in people with depression,” *AAAI Work. Work. Twenty-Eighth AAAI Conf. Artif. Intell.*, pp. 12–16, 2014.

[49] L. Gao, A. Bourke, and N. J, “Activity recognition using dynamic multiple sensor fusion in body sensor networks,” *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, pp. 1077–1080, 2012, doi: 10.1109/EMBC.2012.6346121.

[50] S. Liu, R. X. Gao, D. John, J. W. Staudenmayer, and P. S. Freedson, “Multisensor data fusion for physical activity assessment,” *IEEE Trans. Biomed. Eng.*, vol. 59, no. 3, pp. 687–696, 2012, doi: 10.1109/TBME.2011.2178070.

[51] B. Perriot, J. Argod, J. L. Pepin, and N. Noury, “Characterization of Physical Activity in COPD Patients: Validation of a Robust Algorithm for Actigraphic Measurements in Living Situations,” *IEEE J. Biomed. Heal. Informatics*, vol. 18, no. 4, pp. 1225–1231, 2014, doi: 10.1109/JBHI.2013.2282617.

[52] M. P. Rothney, M. Neumann, A. Beziat, and K. Y. Chen, “An artificial neural network model of energy expenditure using nonintegrated acceleration signals,” *J. Appl. Physiol.*, vol. 103, no. 4, pp. 1419–1427, 2007, doi: 10.1152/japplphysiol.00429.2007.

[53] L. Gao, A. K. Bourke, and J. Nelson, “A comparison of classifiers for activity recognition using multiple accelerometer-based sensors,” *Proc. 11th IEEE Int. Conf. Cybern. Intell. Syst. 2012, CIS 2012*, pp. 149–153, 2012, doi: 10.1109/CIS.2013.6782169.

[54] R. Liu and M. Liu, “Recognizing human activities based on multi-sensors fusion,” *2010 4th Int. Conf. Bioinforma. Biomed. Eng. iCBBE 2010*, pp. 1–4, 2010, doi: 10.1109/ICBBE.2010.5514802.

[55] E. Keogh and S. Chu, “Segmenting Time Series : A Survey and Novel Approach. Data Mining in Time Series Databases,” 2003.

[56] S. Ghose and J. J. Barua, “A systematic approach with data mining for analyzing physical activity for an activity recognition system,” *Proc. 2013 2nd Int. Conf. Adv. Electr. Eng. ICAEE 2013*, no. Icaee, pp. 415–420, 2013, doi: 10.1109/ICAEE.2013.6750374.

[57] E. P. Ijjina and C. K. Mohan, “One-shot periodic activity recognition using convolutional neural networks,” *Proc. - 2014 13th Int. Conf. Mach. Learn. Appl. ICMLA 2014*, pp. 388–391, 2014, doi: 10.1109/ICMLA.2014.69.

[58] S. Yan, Y. Liao, X. Feng, and Y. Liu, “Real time activity recognition on streaming sensor data for smart environments,” *PIC 2016 - Proc. 2016 IEEE Int. Conf. Prog. Informatics Comput.*, pp. 51–55, 2017, doi: 10.1109/PIC.2016.7949466.

[59] J. H. Li, L. Tian, H. Wang, Y. An, K. Wang, and L. Yu, “Segmentation and Recognition of Basic and Transitional Activities for Continuous Physical Human Activity,” *IEEE Access*, vol. 7, pp. 42565–42576, 2019, doi: 10.1109/ACCESS.2019.2905575.

[60] jie wan, M. S. Li, M. O’Grady, X. Gu, M. A.A.H Alawlaqi, and G. O’Hare, “Time-bounded Activity Recognition for Ambient Assisted Living,” *IEEE Trans. Emerg. Top. Comput.*, vol. 6750, no. MAY, pp. 1–13, 2018, doi:

10.1109/TETC.2018.2870047.

[61] L. Chen, C. Nugent, and H. Wang, "A Knowledge-Driven Approach to Activity Recognition in Smart Homes," *Knowl. Data Eng. IEEE Trans.*, vol. PP, no. 99, p. 1, 2011, doi: 10.1109/TKDE.2011.51.

[62] T. Van Kasteren, A. Noulas, G. Englebienne, and B. Kröse, "Accurate activity recognition in a home setting," *UbiComp 2008 - Proc. 10th Int. Conf. Ubiquitous Comput.*, no. July 2014, pp. 1–9, 2008, doi: 10.1145/1409635.1409637.

[63] I. E. Achumba, S. Bersch, R. Khusainov, D. Azzi, and U. Kamalu, "On time series sensor data segmentation for fall and activity classification," *2012 IEEE 14th Int. Conf. e-Health Networking, Appl. Serv. Heal. 2012*, pp. 427–430, 2012, doi: 10.1109/HealthCom.2012.6379453.

[64] M. H. M. Noor, Z. Salcic, and K. I. K. Wang, "Dynamic sliding window method for physical activity recognition using a single tri-axial accelerometer," *Proc. 2015 10th IEEE Conf. Ind. Electron. Appl. ICIEA 2015*, pp. 102–107, 2015, doi: 10.1109/ICIEA.2015.7334092.

[65] S. Kozina, M. Lustrek, and M. Gams, "Dynamic signal segmentation for activity recognition," *Proc. Int. Jt. Conf. Artif. Intell.*, pp. 1–12, 2011, [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=39DA1CCB02A70CBA6D6114EAE071E80A?doi=10.1.1.301.7984&rep=rep1&type=pdf.

[66] N. C. Krishnan and D. J. Cook, "Activity recognition on streaming sensor data," *Pervasive Mob. Comput.*, vol. 10, no. PART B, pp. 138–154, 2014, doi: 10.1016/j.pmcj.2012.07.003.

[67] G. Okeyo, L. Chen, H. Wang, and R. Sterritt, "Dynamic sensor data segmentation for real-time knowledge-driven activity recognition," *Pervasive Mob. Comput.*, vol. 10, no. PART B, pp. 155–172, 2014, doi: 10.1016/j.pmcj.2012.11.004.

[68] N. C. Krishnan, P. Lade, and S. Panchanathan, "Activity gesture spotting using a threshold model based on adaptive boosting," *2010 IEEE Int. Conf. Multimed. Expo, ICME 2010*, pp. 155–160, 2010, doi: 10.1109/ICME.2010.5583013.

[69] H. Junker, O. Amft, P. Lukowicz, and G. Tröster, "Gesture spotting with body-worn inertial sensors to detect user activities," *Pattern Recognit.*, vol. 41, no. 6, pp. 2010–2024, 2008, doi: 10.1016/j.patcog.2007.11.016.

[70] K. Van Laerhoven and E. Berlin, "When else did this happen? Efficient subsequence representation for matching in wearable activity data," *Proc. - Int. Symp. Wearable Comput. ISWC*, pp. 101–104, 2009, doi: 10.1109/ISWC.2009.23.

[71] Z. Wang and Y. Huo, "A multi-Attribute fusion acceleration feature selection algorithm for activity recognition on smart phones," *Proc. - 2014 Int. Conf. Inf. Sci. Electron. Electr. Eng. ISEEE 2014*, vol. 1, pp. 145–148, 2014, doi: 10.1109/InfoSEEE.2014.6948085.

[72] G. He, X. Luan, J. Wang, and X. Wang, "A Novel Feature Map for Human Activity Recognition," *Proc. - 2017 2nd Int. Conf. Mech. Control Comput. Eng. ICMCCE 2017*, vol. 2018-Janua, pp. 216–219, 2018, doi: 10.1109/ICMCCE.2017.47.

[73] X. Guo, X. Hu, X. Ye, C. Hu, C. Song, and H. Wu, "Human Activity Recognition Based on Two-Dimensional Acoustic Arrays," in *IEEE International Ultrasonics Symposium, IUS*, 2018, vol. 2018-Octob, pp. 7–10, doi: 10.1109/ULTSYM.2018.8579884.

[74] O. Dehzangi and V. Sahu, "IMU-Based Robust Human Activity Recognition using

Feature Analysis, Extraction, and Reduction," *Proc. - Int. Conf. Pattern Recognit.*, vol. 2018-Augus, pp. 1402–1407, 2018, doi: 10.1109/ICPR.2018.8546311.

[75] K. Rybina, M. Ternovoy, and W. Dargie, "Expressiveness of time domain features for detecting different types of human movements," *Mod. Probl. Radio Eng. Telecommun. Comput. Sci. - Proc. 10th Int. Conf. TCSET'2010*, p. 98, 2010.

[76] O. P. Gan, "Wearable sensing for activity recognition," *Proc. IECON 2017 - 43rd Annu. Conf. IEEE Ind. Electron. Soc.*, vol. 2017-Janua, pp. 8825–8830, 2017, doi: 10.1109/IECON.2017.8217551.

[77] C. Wang and W. Zhang, "Activity Recognition Based on Smartphone and Dual-Tree Complex Wavelet Transform," *Proc. - 2015 8th Int. Symp. Comput. Intell. Des. Isc. 2015*, vol. 2, pp. 267–270, 2016, doi: 10.1109/ISCID.2015.51.

[78] S. J. Preece, J. Y. Goulermas, L. P. J. Kenney, D. Howard, K. Meijer, and R. Crompton, "Activity identification using body-mounted sensors - A review of classification techniques," *Physiol. Meas.*, vol. 30, no. 4, 2009, doi: 10.1088/0967-3334/30/4/R01.

[79] S. J. Preece, J. Y. Goulermas, L. P. J. Kenney, and D. Howard, "A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data," *IEEE Trans. Biomed. Eng.*, vol. 56, no. 3, pp. 871–879, 2009, doi: 10.1109/TBME.2008.2006190.

[80] T. Zebin, P. J. Scully, and K. B. Ozanyan, "Evaluation of supervised classification algorithms for human activity recognition with inertial sensors," *Proc. IEEE Sensors*, vol. 2017-Decem, pp. 1–3, 2017, doi: 10.1109/ICSENS.2017.8234222.

[81] L. Bao and S. S. Intille, "Activity Recognition from User-Annotated Acceleration Data," *Pervasive Comput.*, pp. 1–17, 2004, doi: 10.1007/b96922.

[82] K. Nakano and B. Chakraborty, "Effect of dynamic feature for human activity recognition using smartphone sensors," *Proc. - 2017 IEEE 8th Int. Conf. Aware. Sci. Technol. iCAST 2017*, vol. 2018-Janua, no. iCAST, pp. 539–543, 2018, doi: 10.1109/ICAwST.2017.8256516.

[83] Z. He, "Activity recognition from accelerometer signals based on wavelet-AR model," *Proc. 2010 IEEE Int. Conf. Prog. Informatics Comput. PIC 2010*, vol. 1, pp. 499–502, 2010, doi: 10.1109/PIC.2010.5687572.

[84] A. Godfrey, R. Conway, M. Leonard, D. Meagher, and G. M. Olaighin, "A Continuous wavelet transform and classification method for delirium motoric subtyping," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 17, no. 3, pp. 298–307, 2009, doi: 10.1109/TNSRE.2009.2023284.

[85] M. Li *et al.*, "Multimodal physical activity recognition by fusing temporal and cepstral information," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 18, no. 4, pp. 369–380, 2010, doi: 10.1109/TNSRE.2010.2053217.

[86] J. Joshi *et al.*, "Multimodal assistive technologies for depression diagnosis and monitoring," *J. Multimodal User Interfaces*, vol. 7, no. 3, pp. 217–228, 2013, doi: 10.1007/s12193-013-0123-2.

[87] T. Pawar, N. S. Anantakrishnan, S. Chaudhuri, and S. P. Duttagupta, "Transition detection in body movement activities for wearable ECG," *IEEE Trans. Biomed. Eng.*, vol. 54, no. 6, pp. 1149–1152, 2007, doi: 10.1109/TBME.2007.891950.

[88] R. Jia and B. Liu, "Human daily activity recognition by fusing accelerometer and multi-lead ECG data," *2013 IEEE Int. Conf. Signal Process. Commun. Comput. ICSPCC*

2013, pp. 1–4, 2013, doi: 10.1109/ICSPCC.2013.6664056.

[89] M. B. Abidine and B. Fergani, “Evaluating C-SVM, CRF and LDA classification for daily activity recognition,” *Proc. 2012 Int. Conf. Multimed. Comput. Syst. ICMCS 2012*, pp. 272–277, 2012, doi: 10.1109/ICMCS.2012.6320300.

[90] A. M. Khan, Y. K. Lee, S. Y. Lee, and T. S. Kim, “A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer,” *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 5, pp. 1166–1172, 2010, doi: 10.1109/TITB.2010.2051955.

[91] S. Kishore, S. Bhattacharjee, and A. Swetapadma, “A hybrid method for activity monitoring using principal component analysis and back-propagation neural network,” pp. 885–889, 2017.

[92] D. Lahiri, C. Dhiman, and D. K. Vishwakarma, “Abnormal human action recognition using average energy images,” *2017 Conf. Inf. Commun. Technol. CICT 2017*, vol. 2018-April, pp. 1–5, 2018, doi: 10.1109/INFOCOMTECH.2017.8340622.

[93] I. Agarwal, A. K. S. Kushwaha, and R. Srivastava, “Weighted fast dynamic time warping based multi-view human activity recognition using a RGB-D sensor,” *2015 5th Natl. Conf. Comput. Vision, Pattern Recognition, Image Process. Graph. NCVPRIPG 2015*, pp. 1–4, 2016, doi: 10.1109/NCVPRIPG.2015.7490046.

[94] Y. Li, D. Xue, E. Forrister, G. Lee, B. Garner, and Y. Kim, “Human Activity Classification Based on Dynamic Time Warping of an On-Body Creeping Wave Signal,” *IEEE Trans. Antennas Propag.*, vol. 64, no. 11, pp. 4901–4905, 2016, doi: 10.1109/TAP.2016.2598199.

[95] H. Foroughi, H. S. Yazdi, H. Pourreza, and M. Javidi, “An eigenspace-based approach for human fall detection using integrated time motion image and multi-class support vector machine,” *Proc. - 2008 IEEE 4th Int. Conf. Intell. Comput. Commun. Process. ICCP 2008*, pp. 83–90, 2008, doi: 10.1109/ICCP.2008.4648358.

[96] A. Murad and J. Y. Pyun, “Deep recurrent neural networks for human activity recognition,” *Sensors (Switzerland)*, vol. 17, no. 11, 2017, doi: 10.3390/s17112556.

[97] J. Azorin-Lopez, M. Saval-Calvo, A. Fuster-Guillo, and J. Garcia-Rodriguez, “Human behaviour recognition based on trajectory analysis using neural networks,” *Proc. Int. Jt. Conf. Neural Networks*, 2013, doi: 10.1109/IJCNN.2013.6706724.

[98] R. Sheng and Z. Zhang, “A hand gesture recognition using single-channel electrodes based on artificial neural network,” no. Itaic, pp. 209–213, 2019, doi: 10.1109/itaic.2019.8785724.

[99] C. Liu, J. Ying, F. Han, and M. Ruan, “Abnormal human activity recognition using bayes classifier and convolutional neural network,” *2018 IEEE 3rd Int. Conf. Signal Image Process. ICSIP 2018*, pp. 33–37, 2019, doi: 10.1109/SIPROCESS.2018.8600483.

[100] H. D. Mehr, H. Polat, and A. Cetin, “Resident activity recognition in smart homes by using artificial neural networks,” *4th Int. Istanbul Smart Grid Congr. Fair, ICSG 2016*, pp. 1–5, 2016, doi: 10.1109/SGCF.2016.7492428.

[101] D. Tian, X. Xu, Y. Tao, and X. Wang, “An improved activity recognition method based on smart watch data,” *Proc. - 2017 IEEE Int. Conf. Comput. Sci. Eng. IEEE/IFIP Int. Conf. Embed. Ubiquitous Comput. CSE EUC 2017*, vol. 1, pp. 756–759, 2017, doi: 10.1109/CSE-EUC.2017.148.

[102] J. Pärkkä, L. Cluitmans, and M. Ermes, "Personalization algorithm for real-time activity recognition using PDA, wireless motion bands, and binary decision tree," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 5, pp. 1211–1215, 2010, doi: 10.1109/TITB.2010.2055060.

[103] Y. C. Huang, C. W. Yi, W. C. Peng, H. C. Lin, and C. Y. Huang, "A study on multiple wearable sensors for activity recognition," *2017 IEEE Conf. Dependable Secur. Comput.*, pp. 449–452, 2017, doi: 10.1109/DESEC.2017.8073827.

[104] Y. J. Hong, I. J. Kim, S. C. Ahn, and H. G. Kim, "Activity recognition using wearable sensors for elder care," *Proc. 2008 2nd Int. Conf. Futur. Gener. Commun. Networking, FGCN 2008*, vol. 2, pp. 302–305, 2008, doi: 10.1109/FGCN.2008.165.

[105] A. Fleury, N. Noury, and M. Vacher, "Introducing knowledge in the process of supervised classification of activities of Daily Living in Health Smart Homes," *12th IEEE Int. Conf. e-Health Networking, Appl. Serv. Heal. 2010*, 2010, doi: 10.1109/HEALTH.2010.5556549.

[106] H. Foroughi, A. Rezvanian, and A. Paziraee, "Robust fall detection using human shape and multi-class support vector machine," *Proc. - 6th Indian Conf. Comput. Vision, Graph. Image Process. ICVGIP 2008*, pp. 413–420, 2008, doi: 10.1109/ICVGIP.2008.49.

[107] L. Xie, J. Tian, G. Ding, and Q. Zhao, "Human activity recognition method based on inertial sensor and barometer," *5th IEEE Int. Symp. Inert. Sensors Syst. Inert. 2018 - Proc.*, pp. 1–4, 2018, doi: 10.1109/ISISS.2018.8358140.

[108] D. N. Tran and D. D. Phan, "Human Activities Recognition in Android Smartphone Using Support Vector Machine," *Proc. - Int. Conf. Intell. Syst. Model. Simulation, ISMS*, pp. 64–68, 2017, doi: 10.1109/ISMS.2016.51.

[109] J. Hossen, E. L. Jacobs, and F. K. Chowdhury, "Activity recognition in thermal infrared video," *Conf. Proc. - IEEE SOUTHEASTCON*, vol. 2015-June, no. June, pp. 1–2, 2015, doi: 10.1109/SECON.2015.7132922.

[110] M. Degirmenci, M. A. Özdemir, R. Sadighzadeh, and A. Akan, "Emotion Recognition from EEG Signals by Using Empirical Mode Decomposition," *2018 Med. Technol. Natl. Congr. TIPTEKNO 2018*, pp. 1–4, 2018, doi: 10.1109/TIPTEKNO.2018.8597061.

[111] X. Yang, A. Dinh, and L. Chen, "Implementation of a wearerable real-time system for physical activity recognition based on naive bayes classifier," *ICBBT 2010 - 2010 Int. Conf. Bioinforma. Biomed. Technol.*, pp. 101–105, 2010, doi: 10.1109/ICBBT.2010.5479000.

[112] P. Gupta and T. Dallas, "Feature selection and activity recognition system using a single triaxial accelerometer," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 6, pp. 1780–1786, 2014, doi: 10.1109/TBME.2014.2307069.

[113] M. K. Hasan, H. A. Rubaiyeat, Y. K. Lee, and S. Lee, "A reconfigurable HMM for activity recognition," *Int. Conf. Adv. Commun. Technol. ICACT*, vol. 1, pp. 843–846, 2008, doi: 10.1109/ICACT.2008.4493886.

[114] Y. Wang, K. Huang, and T. Tan, "Abnormal Activity Recognition in Office Based on R Transform," pp. 209–212, 2007.

[115] Z. Uddin, S. Member, N. D. Thang, and T. Kim, "Human Activity Recognition via 3-D Joint Angle Features and Hidden Markov Models," *Training*, pp. 713–716, 2010, doi: 10.1109/ICIP.2010.5651953.

[116] A. C. Tsai, Y. Y. Ou, C. A. Sun, and J. F. Wang, "VQ-HMM classifier for human activity recognition based on R-GBD sensor," *Proc. 2017 Int. Conf. Orange Technol. ICOT 2017*, vol. 2018-Janua, pp. 201–204, 2018, doi: 10.1109/ICOT.2017.8336122.

[117] S. Agatonovic-Kustrin and R. Beresford, "Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research," *J. Pharm. Biomed. Anal.*, vol. 22, no. 5, pp. 717–727, 2000, doi: 10.1016/S0731-7085(99)00272-1.

[118] G. P. Zhang, "Neural networks for classification: A survey," *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 30, no. 4, pp. 451–462, 2000, doi: 10.1109/5326.897072.

[119] A. Navada, A. N. Ansari, S. Patil, and B. A. Sonkamble, "Overview of use of decision tree algorithms in machine learning," *Proc. - 2011 IEEE Control Syst. Grad. Res. Colloquium, ICSGRC 2011*, pp. 37–42, 2011, doi: 10.1109/ICSGRC.2011.5991826.

[120] L. Rokach and O. Maimon, "Top-Down Induction of Decision Trees," vol. 35, no. 4, pp. 476–487, 2005.

[121] S. Rasoul and L. David, "A Survey of Decision Tree Classifier Methodology," *IEEE Trans. Syst. Man. Cybern.*, vol. 21, no. 3, pp. 660–674, 1991.

[122] C. M. Patil, B. Jagadeesh, and M. N. Meghana, "An Approach of Understanding Human Activity Recognition and Detection for Video Surveillance using HOG Descriptor and SVM Classifier," *Int. Conf. Curr. Trends Comput. Electr. Electron. Commun. CTCEEC 2017*, pp. 481–485, 2018, doi: 10.1109/CTCEEC.2017.8455046.

[123] H. Byun and S. W. Lee, "Applications of support vector machines for pattern recognition: A survey," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 2388, pp. 213–236, 2002, doi: 10.1007/3-540-45665-1_17.

[124] A. Pradhan, "Support vector machine-A survey," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 2, no. 8, 2012.

[125] L. Gao, A. K. Bourke, and J. Nelson, "An efficient sensing approach using dynamic multi-sensor collaboration for activity recognition," *2011 Int. Conf. Distrib. Comput. Sens. Syst. Work. DCOSS'11*, pp. 1–3, 2011, doi: 10.1109/DCOSS.2011.5982190.

[126] N. Alshurafa *et al.*, "Designing a Robust Activity Recognition Framework for Health and Exergaming Using Wearable Sensors," *IEEE J. Biomed. Heal. Informatics*, vol. 18, no. 5, pp. 1636–1646, 2014, doi: 10.1109/JBHI.2013.2287504.

[127] H. L. Cardoso and J. M. Moreira, "Human Activity Recognition by Means of Online Semi-supervised Learning," pp. 75–77, 2016, doi: 10.1109/mdm.2016.93.

[128] M. Lv, L. Chen, T. Chen, and G. Chen, "Bi-View Semi-Supervised Learning Based Semantic Human Activity Recognition Using Accelerometers," *IEEE Trans. Mob. Comput.*, vol. 17, no. 9, pp. 1991–2001, 2018, doi: 10.1109/TMC.2018.2793913.

[129] M. Zeng, T. Yu, X. Wang, L. T. Nguyen, O. J. Mengshoel, and I. Lane, "Semi-supervised convolutional neural networks for human activity recognition," *Proc. - 2017 IEEE Int. Conf. Big Data, Big Data 2017*, vol. 2018-Janua, pp. 522–529, 2018, doi: 10.1109/BigData.2017.8257967.

[130] K. Norton, L. Norton, and D. Sadgrove, "Position statement on physical activity and exercise intensity terminology," *J. Sci. Med. Sport*, vol. 13, no. 5, pp. 496–502, 2010, doi: 10.1016/j.jsams.2009.09.008.

[131] F. Corno, L. De Russis, and A. Monge Roffarello, "A Healthcare Support System

for Assisted Living Facilities: An IoT Solution," *Proc. - Int. Comput. Softw. Appl. Conf.*, vol. 1, pp. 344–352, 2016, doi: 10.1109/COMPSAC.2016.29.

[132] N. E. Tabbakha, W. H. Tan, and C. P. Ooi, "Indoor location and motion tracking system for elderly assisted living home," *Proceeding 2017 Int. Conf. Robot. Autom. Sci. ICORAS 2017*, vol. 2018-March, pp. 1–4, 2018, doi: 10.1109/ICORAS.2017.8308073.

[133] L. Xu and N. Pombo, "Human Behavior Prediction Though Noninvasive and Privacy-Preserving Internet of Things (IoT) Assisted Monitoring," *2019 IEEE 5th World Forum Internet Things*, pp. 773–777, 2019, doi: 10.1109/wf-iot.2019.8767301.

[134] S. Greene, H. Thapliyal, and D. Carpenter, "IoT-Based fall detection for smart home environments," *Proc. - 2016 IEEE Int. Symp. Nanoelectron. Inf. Syst. iNIS 2016*, pp. 23–28, 2017, doi: 10.1109/iNIS.2016.017.

[135] M. Marzencki, P. Lin, T. Cho, J. Guo, B. Ngai, and B. Kaminska, "Remote health, activity, and asset monitoring with wireless sensor networks," *2011 IEEE 13th Int. Conf. e-Health Networking, Appl. Serv. Heal. 2011*, pp. 98–101, 2011, doi: 10.1109/HEALTH.2011.6026796.

[136] T. Bratan and M. Clarke, "Optimum design of remote patient monitoring systems," *Annu. Int. Conf. IEEE Eng. Med. Biol. - Proc.*, pp. 6465–6468, 2006, doi: 10.1109/IEMBS.2006.260729.

[137] E. Baba, A. Jilbab, and A. Hammouch, "A health remote monitoring application based on wireless body area networks," *2018 Int. Conf. Intell. Syst. Comput. Vision, ISCV 2018*, vol. 2018-May, pp. 1–4, 2018, doi: 10.1109/ISACV.2018.8354042.

[138] N. Georgi and R. Le Bouquin Jeannes, "Proposal of a remote monitoring system for elderly health prevention," *2017 Int. Conf. Smart, Monit. Control. Cities, SM2C 2017*, pp. 69–74, 2017, doi: 10.1109/SM2C.2017.8071261.

[139] U. Varshney, "A framework for wireless monitoring of mental health conditions," *Proc. 31st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. Eng. Futur. Biomed. EMBC 2009*, pp. 5219–5222, 2009, doi: 10.1109/IEMBS.2009.5334284.

[140] Y. Honda, H. Yamaguchi, and T. Higashino, "Daily Activity Recognition based on Markov Logic Network for Elderly Monitoring," *2019 16th IEEE Annu. Consum. Commun. Netw. Conf. CCNC 2019*, pp. 1–6, 2019, doi: 10.1109/CCNC.2019.8651817.

[141] H. Xu, Y. Pan, J. Li, L. Nie, and X. Xu, "Activity Recognition Method for Home-Based Elderly Care Service Based on Random Forest and Activity Similarity," *IEEE Access*, vol. 7, pp. 16217–16225, 2019, doi: 10.1109/ACCESS.2019.2894184.

[142] L. Newcombe, P. Yang, C. Cater, M. Hanneghan, and J. Qi, "Experimental Analysis of Cost-Effective Mobile Sensing Technologies for Activity Analytics in Elderly Care," *Proc. - 20th Int. Conf. High Perform. Comput. Commun. 16th Int. Conf. Smart City 4th Int. Conf. Data Sci. Syst. HPCC/SmartCity/DSS 2018*, pp. 1442–1448, 2019, doi: 10.1109/HPCC/SmartCity/DSS.2018.00238.

[143] L. Atallah, B. Lo, R. King, and G. Z. Yang, "Sensor positioning for activity recognition using wearable accelerometers," *IEEE Trans. Biomed. Circuits Syst.*, vol. 5, no. 4, pp. 320–329, 2011, doi: 10.1109/TBCAS.2011.2160540.

[144] H. Ghayvat, S. Mukhopadhyay, B. Shenjie, A. Chouhan, and W. Chen, "Smart home based ambient assisted living: Recognition of anomaly in the activity of daily living for an elderly living alone," *I2MTC 2018 - 2018 IEEE Int. Instrum. Meas. Technol. Conf. Discov. New Horizons Instrum. Meas. Proc.*, pp. 1–5, 2018, doi:

10.1109/I2MTC.2018.8409885.

[145] L. Newcombe, P. Yang, C. Carter, and M. Hanneghan, "Internet of Things Enabled Technologies for Behaviour Analytics in Elderly Person Care: A Survey," *2017 IEEE Int. Conf. Internet Things IEEE Green Comput. Commun. IEEE Cyber, Phys. Soc. Comput. IEEE Smart Data*, pp. 863–870, 2017, doi: 10.1109/iThings-GreenCom-CPSCom-SmartData.2017.133.

[146] P. Anastasopoulou, M. Tansella, J. Stumpp, L. Shammas, and S. Hey, "Classification of human physical activity and energy expenditure estimation by accelerometry and barometry," *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, pp. 6451–6454, 2012, doi: 10.1109/EMBC.2012.6347471.

[147] S. Liu, R. X. Gao, D. John, J. W. Staudenmayer, and P. S. Freedson, "Multisensor data fusion for physical activity assessment," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 3, pp. 687–696, 2012, doi: 10.1109/TBME.2011.2178070.

[148] E. Kim, S. Helal, and D. Cook, "Human activity recognition and pattern discovery," *IEEE Pervasive Comput.*, vol. 9, no. 1, pp. 48–53, 2010, doi: 10.1109/MPRV.2010.7.

[149] M. Kheirkhahan *et al.*, "A bag-of-words approach for assessing activities of daily living using wrist accelerometer data," *Proc. - 2017 IEEE Int. Conf. Bioinforma. Biomed. BIBM 2017*, vol. 2017-Janua, pp. 678–685, 2017, doi: 10.1109/BIBM.2017.8217735.

[150] R. Ali, L. Atallah, B. Lo, and G. Z. Yang, "Detection and analysis of transitional activity in manifold space," *IEEE Trans. Inf. Technol. Biomed.*, vol. 16, no. 1, pp. 119–128, 2012, doi: 10.1109/TITB.2011.2165320.

[151] S. Okour, A. Maeder, and J. Basilakis, "An Adaptive Rule-Based Approach to Classifying Activities of Daily Living," *Proc. - 2015 IEEE Int. Conf. Healthc. Informatics, ICHI 2015*, pp. 404–407, 2015, doi: 10.1109/ICHI.2015.57.

[152] A. Das Antar, M. Ahmed, and M. A. R. Ahad, "Challenges in sensor-based human activity recognition and a comparative analysis of benchmark datasets: A review," *2019 Jt. 8th Int. Conf. Informatics, Electron. Vision, ICIEV 2019 3rd Int. Conf. Imaging, Vis. Pattern Recognition, icIVPR 2019 with Int. Conf. Act. Behav. Comput. ABC 2019*, pp. 134–139, 2019, doi: 10.1109/ICIEV.2019.8858508.

[153] D. Ferretti, E. Principi, S. Squartini, and L. Mandolini, "An experimental study on new features for activity of daily living recognition," *Proc. Int. Jt. Conf. Neural Networks*, vol. 2016-Octob, pp. 3958–3965, 2016, doi: 10.1109/IJCNN.2016.7727713.

[154] A. Allik, K. Pilt, D. Karai, I. Fridolin, M. Leier, and G. Jervan, "Activity classification for real-Time wearable systems: Effect of window length, sampling frequency and number of features on classifier performance," *IECBES 2016 - IEEE-EMBS Conf. Biomed. Eng. Sci.*, pp. 460–464, 2016, doi: 10.1109/IECBES.2016.7843493.

[155] J. E. Sasaki, A. Hickey, J. Staudenmayer, D. John, J. A. Kent, and P. S. Freedson, "Performance of Activity Classification Algorithms in Free-living Older Adults," *Physiol. Behav.*, vol. 176, no. 1, pp. 139–148, 2017, doi: 10.1249/MSS.0000000000000844.

[156] J. Staudenmayer, D. Pober, S. Crouter, D. Bassett, and P. Freedson, "An artificial neural network to estimate physical activity energy expenditure and identify physical activity type from an accelerometer," *J. Appl. Physiol.*, vol. 107, no. 4, pp. 1300–1307, 2009, doi: 10.1152/japplphysiol.00465.2009.

[157] S. Zhang, A. V. Rowlands, P. Murray, and T. L. Hurst, "Physical activity classification using the GENEA wrist-worn accelerometer," *Med. Sci. Sports Exerc.*,

vol. 44, no. 4, pp. 742–748, 2012, doi: 10.1249/MSS.0b013e31823bf95c.

[158] M. S. Totty and E. Wade, "Muscle Activation and Inertial Motion Data for Noninvasive Classification of Activities of Daily Living," *IEEE Trans. Biomed. Eng.*, vol. 65, no. 5, pp. 1069–1076, 2018, doi: 10.1109/TBME.2017.2738440.

[159] A. Z. Rakhman, L. E. Nugroho, Widyawan, and Kurnianingsih, "Fall detection system using accelerometer and gyroscope based on smartphone," *2014 1st Int. Conf. Inf. Technol. Comput. Electr. Eng. Green Technol. Its Appl. a Better Futur. ICITACEE 2014 - Proc.*, pp. 99–104, 2015, doi: 10.1109/ICITACEE.2014.7065722.

[160] E. Kantoch, P. Augustyniak, M. Markiewicz, and D. Prusak, "Monitoring activities of daily living based on wearable wireless body sensor network," *2014 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBC 2014*, pp. 586–589, 2014, doi: 10.1109/EMBC.2014.6943659.

[161] J. A. Ward, P. Lukowicz, G. Tröster, and T. E. Starner, "Activity recognition of assembly tasks using body-worn microphones and accelerometers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1553–1566, 2006, doi: 10.1109/TPAMI.2006.197.

[162] B. Das, B. L. Thomas, A. M. Seelye, D. J. Cook, L. B. Holder, and M. Schmitter-Edgecombe, "Context-aware prompting from your smart phone," *2012 IEEE Consum. Commun. Netw. Conf. CCNC'2012*, pp. 56–57, 2012, doi: 10.1109/CCNC.2012.6181049.

[163] I. Orha and S. Oniga, "Study regarding the optimal sensors placement on the body for human activity recognition," *2014 IEEE 20th Int. Symp. Des. Technol. Electron. Packag. SIITME 2014*, pp. 203–206, 2014, doi: 10.1109/SIITME.2014.6967028.

[164] U. Maurer, a. Smailagic, D. P. Siewiorek, and M. Deisher, "Activity recognition and monitoring using multiple sensors on different body positions," *Int. Work. Wearable Implant. Body Sens. Networks*, pp. 4–7, 2006, doi: 10.1109/BSN.2006.6.

[165] M. J. Mathie, A. C. F. Coster, N. H. Lovell, and B. G. Celler, "Detection of daily physical activities using a triaxial accelerometer," *Med. Biol. Eng. Comput.*, vol. 41, no. 3, pp. 296–301, 2003, doi: 10.1007/BF02348434.

[166] C. Wang, J. Zhang, Z. Wang, and J. Wang, "Position-independent activity recognition model for smartphone based on frequency domain algorithm," *Proc. 2013 3rd Int. Conf. Comput. Sci. Netw. Technol. ICCSNT 2013*, pp. 396–399, 2014, doi: 10.1109/ICCSNT.2013.6967138.

[167] P. Esfahani and H. T. Malazi, "PAMS: A new position-aware multi-sensor dataset for human activity recognition using smartphones," *2017 19th Int. Symp. Comput. Archit. Digit. Syst. CADS 2017*, vol. 2018-Janua, pp. 1–7, 2018, doi: 10.1109/CADS.2017.8310680.

[168] H. Rezaie and M. Ghassemian, "An Adaptive Algorithm to Improve Energy Efficiency in Wearable Activity Recognition Systems," *IEEE Sens. J.*, vol. 17, no. 16, pp. 5315–5323, 2017, doi: 10.1109/JSEN.2017.2720725.

[169] X. Qi, M. Keally, G. Zhou, Y. Li, and Z. Ren, "AdaSense: Adapting sampling rates for activity recognition in Body Sensor Networks," *Real-Time Technol. Appl. - Proc.*, pp. 163–172, 2013, doi: 10.1109/RTAS.2013.6531089.

[170] A. Nk, G. Bhat, J. Park, H. G. Lee, and U. Y. Ogras, "Sensor-classifier co-optimization for wearable human activity recognition applications," *2019 IEEE Int. Conf. Embed. Softw. Syst. ICESS 2019*, pp. 1–4, 2019, doi: 10.1109/ICESS.2019.8782506.

[171] "Shimmer: Discovery in Motion." http://www.shimmersensing.com (accessed Oct. 15, 2018).

[172] N. Yan, J. Chen, and T. Yu, "A Feature Set for the Similar Activity Recognition Using Smartphone," *2018 10th Int. Conf. Wirel. Commun. Signal Process. WCSP 2018*, pp. 1–6, 2018, doi: 10.1109/WCSP.2018.8555704.

[173] P. Sarcevic, S. Pletl, and Z. Kincses, "Comparison of time- and frequency-domain features for movement classification using data from wrist-worn sensors," *SISY 2017 - IEEE 15th Int. Symp. Intell. Syst. Informatics, Proc.*, pp. 261–265, 2017, doi: 10.1109/SISY.2017.8080564.

[174] E. D. Nugroho, M. E. Wibowo, and R. Pulungan, "Parallel Implementation of Genetic Algorithm for Searching Optimal Parameters of Artificial Neural Networks," pp. 0–5, 2017.

[175] M. Ussath, D. Jaeger, F. Cheng, and C. Meinel, "Identifying Suspicious User Behavior with Neural Networks," *Proc. - 4th IEEE Int. Conf. Cyber Secur. Cloud Comput. CSCloud 2017 3rd IEEE Int. Conf. Scalable Smart Cloud, SSC 2017*, pp. 255–263, 2017, doi: 10.1109/CSCloud.2017.10.

[176] M. S. Astriani, Y. Heryadi, G. P. Kusuma, and E. Abdurachman, "Smartphone sensors selection using decision tree and KNN to detect head movements in virtual reality application," *Proc. - 2017 Int. Conf. Appl. Comput. Commun. Technol. ComCom 2017*, vol. 2017-Janua, pp. 1–5, 2017, doi: 10.1109/COMCOM.2017.8167101.

[177] R. Khusainov, D. Azzi, I. E. Achumba, and S. D. Bersch, "Real-time human ambulation, activity, and physiological monitoring: Taxonomy of issues, techniques, applications, challenges and limitations," *Sensors (Switzerland)*, vol. 13, no. 10, pp. 12852–12902, 2013, doi: 10.3390/s131012852.

[178] A. Savitzky and M. J. E. Golay, "Smoothing and Differentiation of Data by Simplified Least Squares Procedures," *Anal. Chem.*, vol. 36, no. 8, pp. 1627–1639, 1964, doi: 10.1021/ac60214a047.

[179] Tensorflow, "Tensorflow." .

[180] Keras, "Keras." .

[181] W. Xiang, H.-D. Tran, and T. T. Johnson, "Reachable Set Computation and Safety Verification for Neural Networks with ReLU Activations," pp. 1–19, 2017, [Online]. Available: http://arxiv.org/abs/1712.08163.

[182] L. Liu, S. Yang, and D. Shi, "Advanced Convolutional Neural Network with Feedforward Inhibition," *Proc. - Int. Conf. Mach. Learn. Cybern.*, vol. 2019-July, pp. 1–5, 2019, doi: 10.1109/ICMLC48188.2019.8949229.

[183] Y. Guo, L. Sun, Z. Zhang, and H. He, "Algorithm Research on Improving Activation Function of Convolutional Neural Networks," *Proc. 31st Chinese Control Decis. Conf. CCDC 2019*, pp. 3582–3586, 2019, doi: 10.1109/CCDC.2019.8833156.

[184] A. F. Agarap, "Deep Learning using Rectified Linear Units (ReLU)," no. 1, pp. 2–8, 2018, [Online]. Available: http://arxiv.org/abs/1803.08375.

[185] Y. Ho and S. Wookey, "The Real-World-Weight Cross-Entropy Loss Function: Modeling the Costs of Mislabeling," *IEEE Access*, vol. 8, pp. 4806–4813, 2020, doi: 10.1109/ACCESS.2019.2962617.

[186] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.

[187] "Join Dementia Research." https://www.joindementiaresearch.nihr.ac.uk/ (accessed Jan. 08, 2019).

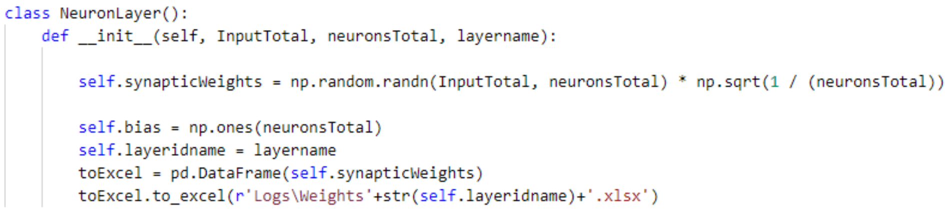
[188] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Syntethic Minority Over-sampling Technique," *J. Artif. Intell. Res.*, vol. 2009, no. Sept. 28, pp. 321–357, 2006, doi: 10.1613/jair.953.

Appendices

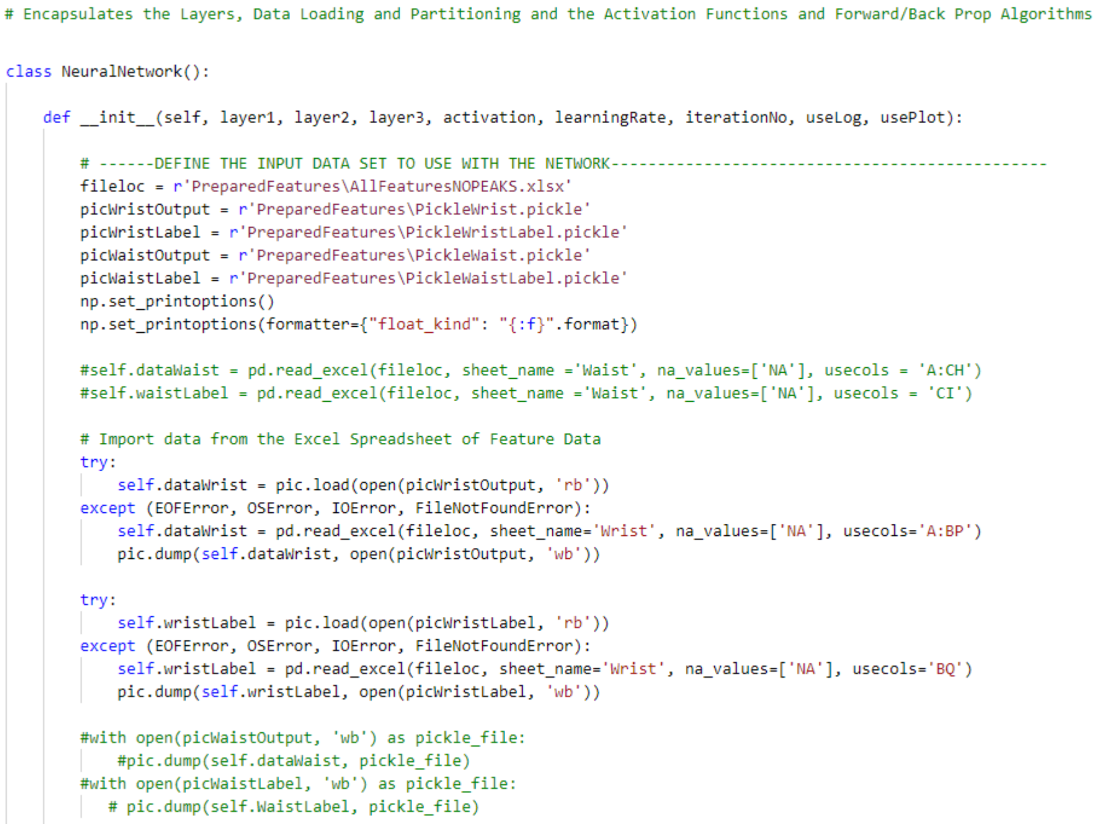
Appendix A – Prototype Artificial Neural Network

In order to develop the base implementation of the ANN and determine the neural connections required, a Python-based prototype was developed.

Layer and Network Models:

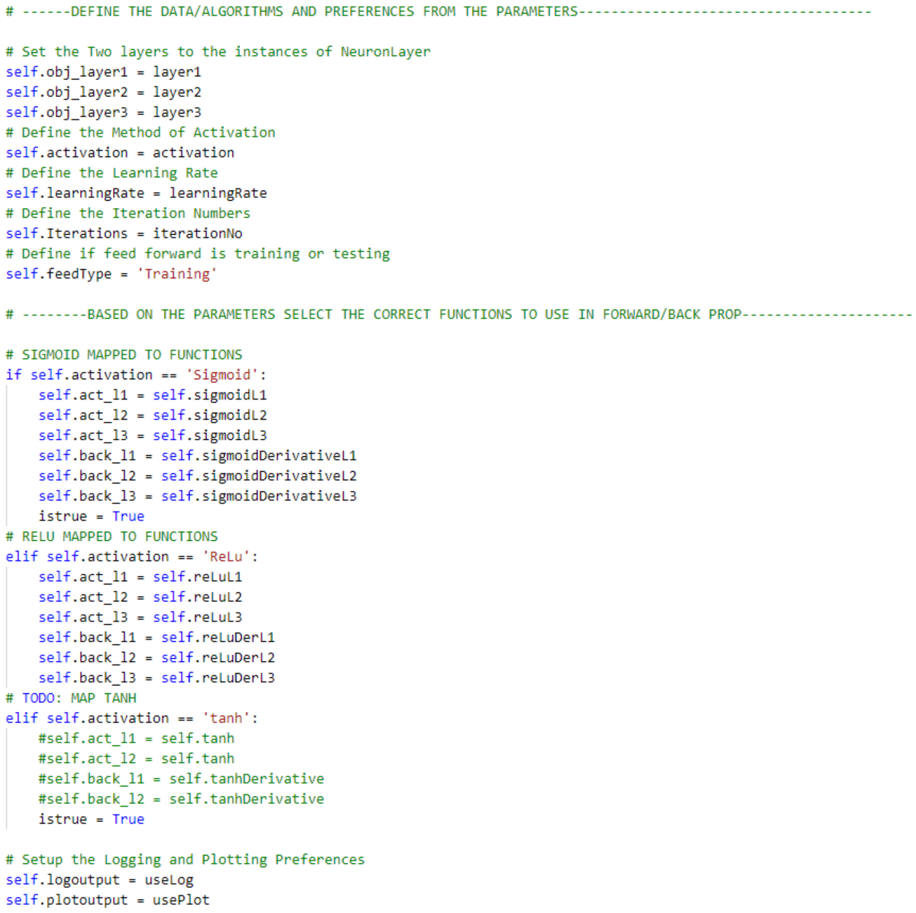


```
class NeuronLayer():
    def __init__(self, InputTotal, neuronsTotal, layername):

        self.synapticWeights = np.random.randn(InputTotal, neuronsTotal) * np.sqrt(1 / (neuronsTotal))

        self.bias = np.ones(neuronsTotal)
        self.layeridname = layername
        toExcel = pd.DataFrame(self.synapticWeights)
        toExcel.to_excel(r'Logs\Weights'+str(self.layeridname)+'.xlsx')
```



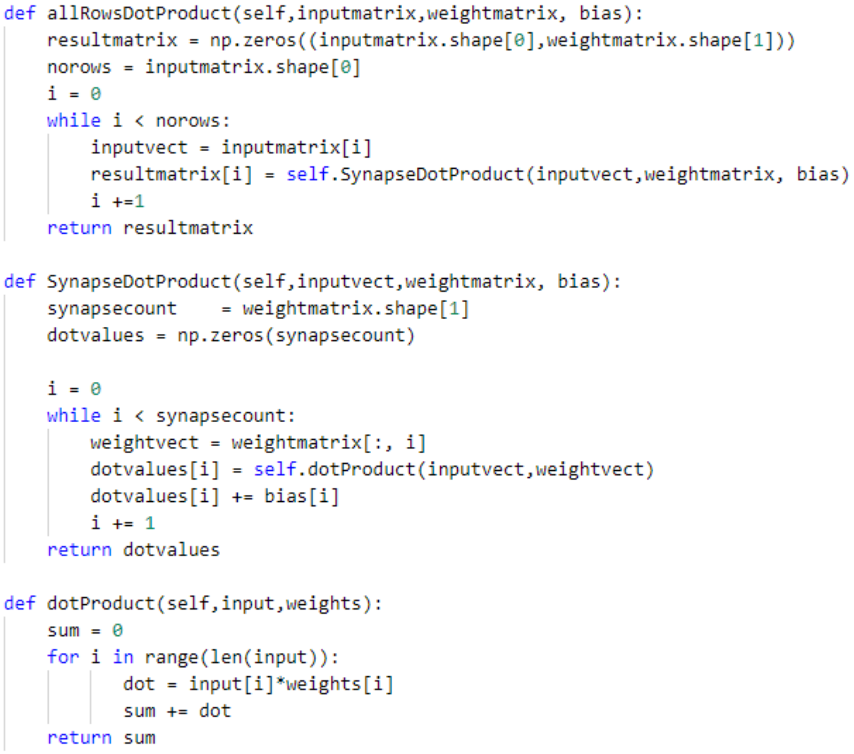
```
# Encapsulates the Layers, Data Loading and Partitioning and the Activation Functions and Forward/Back Prop Algorithms


class NeuralNetwork():

    def __init__(self, layer1, layer2, layer3, activation, learningRate, iterationNo, useLog, usePlot):

        # ------DEFINE THE INPUT DATA SET TO USE WITH THE NETWORK----------------------------------------------
        fileloc = r'PreparedFeatures\AllFeaturesNOPEAKS.xlsx'
        picWristOutput = r'PreparedFeatures\PickleWrist.pickle'
        picWristLabel = r'PreparedFeatures\PickleWristLabel.pickle'
        picWaistOutput = r'PreparedFeatures\PickleWaist.pickle'
        picWaistLabel = r'PreparedFeatures\PickleWaistLabel.pickle'
        np.set_printoptions()
        np.set_printoptions(formatter={"float_kind": "{:f}".format})

        #self.dataWaist = pd.read_excel(fileloc, sheet_name ='Waist', na_values=['NA'], usecols = 'A:CH')
        #self.waistLabel = pd.read_excel(fileloc, sheet_name ='Waist', na_values=['NA'], usecols = 'CI')

        # Import data from the Excel Spreadsheet of Feature Data
        try:
            self.dataWrist = pic.load(open(picWristOutput, 'rb'))
        except (EOFError, OSError, IOError, FileNotFoundError):
            self.dataWrist = pd.read_excel(fileloc, sheet_name='Wrist', na_values=['NA'], usecols='A:BP')
            pic.dump(self.dataWrist, open(picWristOutput, 'wb'))

        try:
            self.wristLabel = pic.load(open(picWristLabel, 'rb'))
        except (EOFError, OSError, IOError, FileNotFoundError):
            self.wristLabel = pd.read_excel(fileloc, sheet_name='Wrist', na_values=['NA'], usecols='BQ')
            pic.dump(self.wristLabel, open(picWristLabel, 'wb'))

        #with open(picWaistOutput, 'wb') as pickle_file:
            #pic.dump(self.dataWaist, pickle_file)
        #with open(picWaistLabel, 'wb') as pickle_file:
           # pic.dump(self.WaistLabel, pickle_file)
```

Layer Configuration



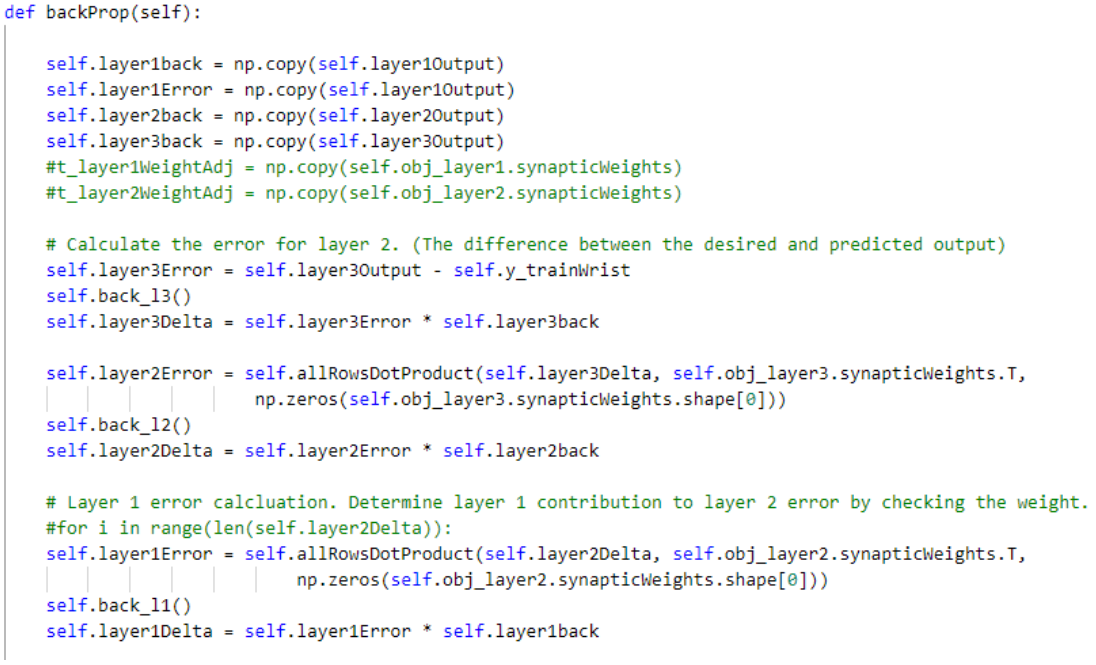
```
# ------DEFINE THE DATA/ALGORITHMS AND PREFERENCES FROM THE PARAMETERS-----------------------------------

# Set the Two layers to the instances of NeuronLayer
self.obj_layer1 = layer1
self.obj_layer2 = layer2
self.obj_layer3 = layer3
# Define the Method of Activation
self.activation = activation
# Define the Learning Rate
self.learningRate = learningRate
# Define the Iteration Numbers
self.Iterations = iterationNo
# Define if feed forward is training or testing
self.feedType = 'Training'

# --------BASED ON THE PARAMETERS SELECT THE CORRECT FUNCTIONS TO USE IN FORWARD/BACK PROP--------------------

# SIGMOID MAPPED TO FUNCTIONS
if self.activation == 'Sigmoid':
    self.act_l1 = self.sigmoidL1
    self.act_l2 = self.sigmoidL2
    self.act_l3 = self.sigmoidL3
    self.back_l1 = self.sigmoidDerivativeL1
    self.back_l2 = self.sigmoidDerivativeL2
    self.back_l3 = self.sigmoidDerivativeL3
    istrue = True
# RELU MAPPED TO FUNCTIONS
elif self.activation == 'ReLu':
    self.act_l1 = self.reLuL1
    self.act_l2 = self.reLuL2
    self.act_l3 = self.reLuL3
    self.back_l1 = self.reLuDerL1
    self.back_l2 = self.reLuDerL2
    self.back_l3 = self.reLuDerL3
# TODO: MAP TANH
elif self.activation == 'tanh':
    #self.act_l1 = self.tanh
    #self.act_l2 = self.tanh
    #self.back_l1 = self.tanhDerivative
    #self.back_l2 = self.tanhDerivative
    istrue = True

# Setup the Logging and Plotting Preferences
self.logoutput = useLog
self.plotoutput = usePlot
```

Synapse Calculation:

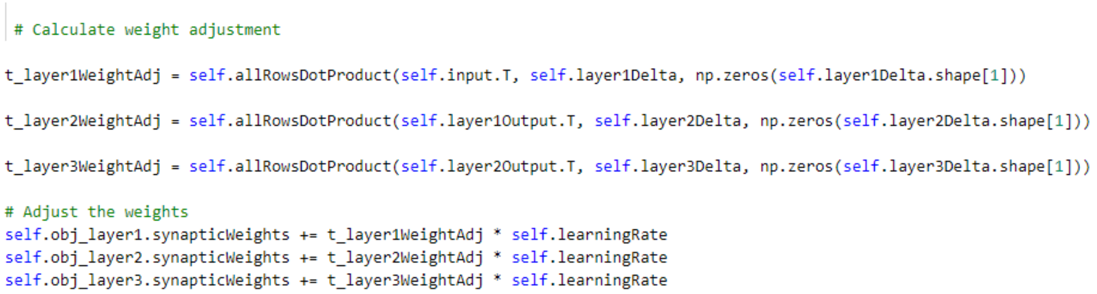


```
def allRowsDotProduct(self,inputmatrix,weightmatrix, bias):
    resultmatrix = np.zeros((inputmatrix.shape[0],weightmatrix.shape[1]))
    norows = inputmatrix.shape[0]
    i = 0
    while i < norows:
        inputvect = inputmatrix[i]
        resultmatrix[i] = self.SynapseDotProduct(inputvect,weightmatrix, bias)
        i +=1
    return resultmatrix

def SynapseDotProduct(self,inputvect,weightmatrix, bias):
    synapsecount    = weightmatrix.shape[1]
    dotvalues = np.zeros(synapsecount)

    i = 0
    while i < synapsecount:
        weightvect = weightmatrix[:, i]
        dotvalues[i] = self.dotProduct(inputvect,weightvect)
        dotvalues[i] += bias[i]
        i += 1
    return dotvalues

def dotProduct(self,input,weights):
    sum = 0
    for i in range(len(input)):
            dot = input[i]*weights[i]
            sum += dot
    return sum
```

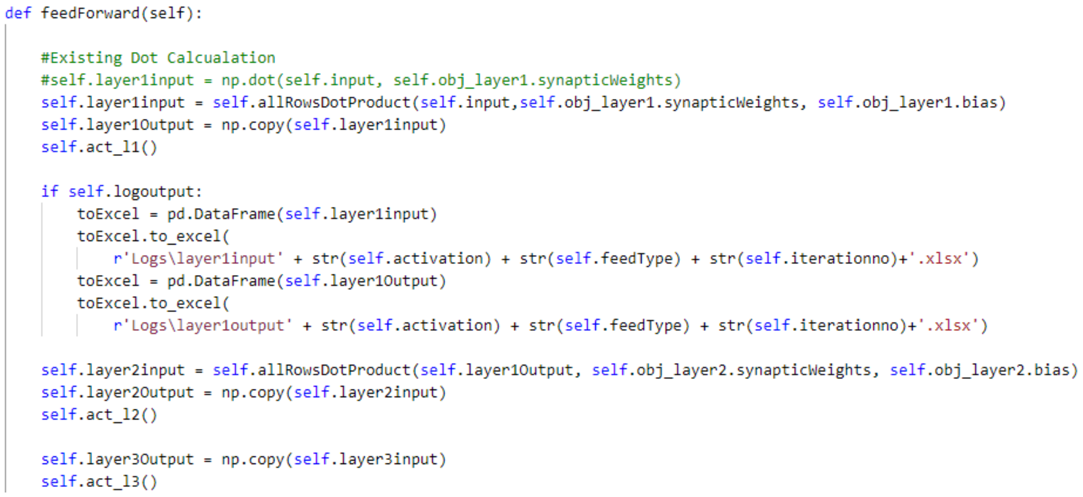
Back Propagation:



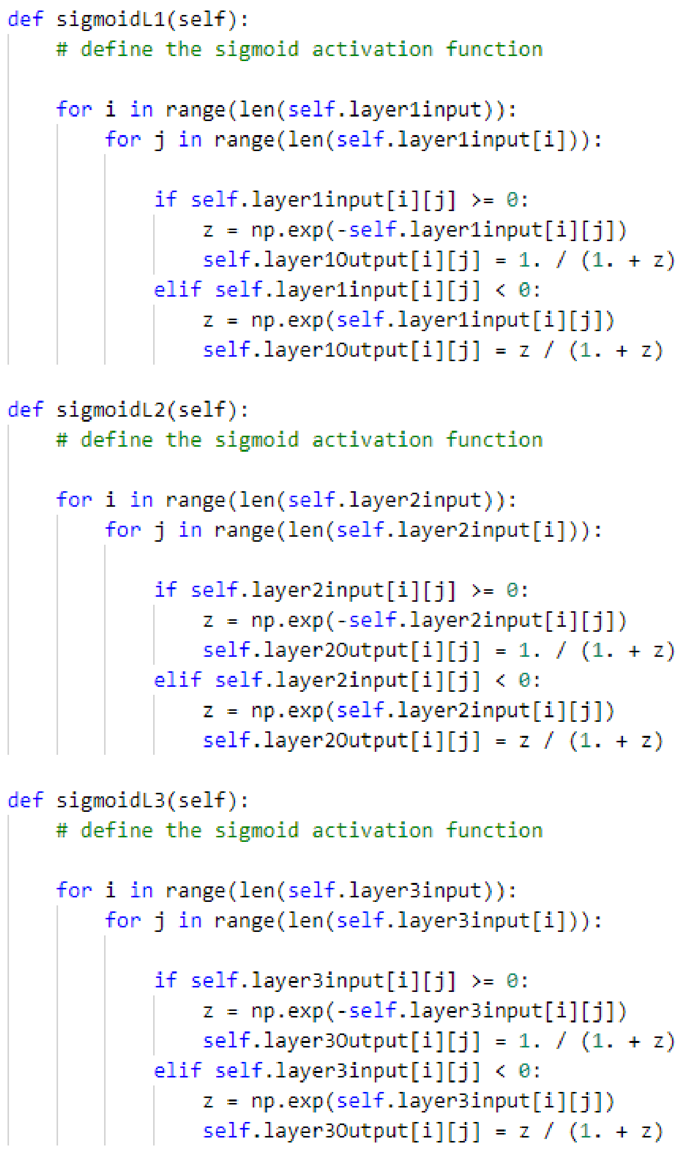
```
def backProp(self):

    self.layer1back = np.copy(self.layer1Output)
    self.layer1Error = np.copy(self.layer1Output)
    self.layer2back = np.copy(self.layer2Output)
    self.layer3back = np.copy(self.layer3Output)
    #t_layer1WeightAdj = np.copy(self.obj_layer1.synapticWeights)
    #t_layer2WeightAdj = np.copy(self.obj_layer2.synapticWeights)

    # Calculate the error for layer 2. (The difference between the desired and predicted output)
    self.layer3Error = self.layer3Output - self.y_trainWrist
    self.back_l3()
    self.layer3Delta = self.layer3Error * self.layer3back

    self.layer2Error = self.allRowsDotProduct(self.layer3Delta, self.obj_layer3.synapticWeights.T,
                         np.zeros(self.obj_layer3.synapticWeights.shape[0]))
    self.back_l2()
    self.layer2Delta = self.layer2Error * self.layer2back

    # Layer 1 error calcluation. Determine layer 1 contribution to layer 2 error by checking the weight.
    #for i in range(len(self.layer2Delta)):
    self.layer1Error = self.allRowsDotProduct(self.layer2Delta, self.obj_layer2.synapticWeights.T,
                             np.zeros(self.obj_layer2.synapticWeights.shape[0]))
    self.back_l1()
    self.layer1Delta = self.layer1Error * self.layer1back
```



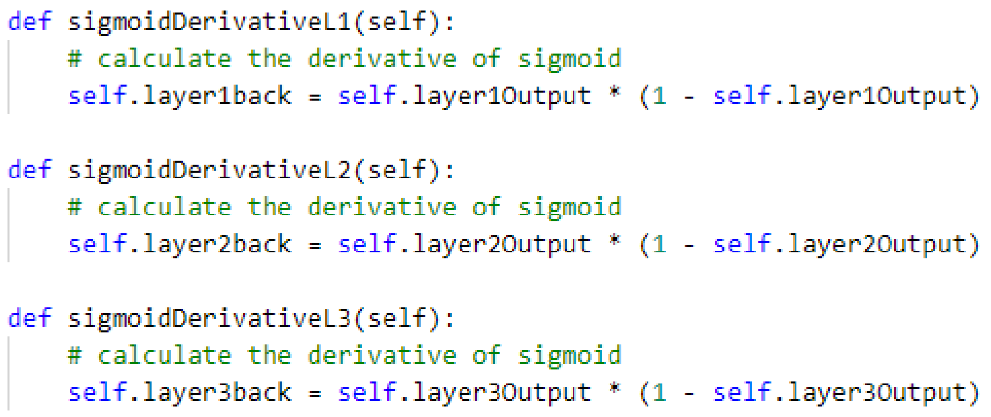
```
# Calculate weight adjustment

t_layer1WeightAdj = self.allRowsDotProduct(self.input.T, self.layer1Delta, np.zeros(self.layer1Delta.shape[1]))

t_layer2WeightAdj = self.allRowsDotProduct(self.layer1Output.T, self.layer2Delta, np.zeros(self.layer2Delta.shape[1]))

t_layer3WeightAdj = self.allRowsDotProduct(self.layer2Output.T, self.layer3Delta, np.zeros(self.layer3Delta.shape[1]))

# Adjust the weights
self.obj_layer1.synapticWeights += t_layer1WeightAdj * self.learningRate
self.obj_layer2.synapticWeights += t_layer2WeightAdj * self.learningRate
self.obj_layer3.synapticWeights += t_layer3WeightAdj * self.learningRate
```



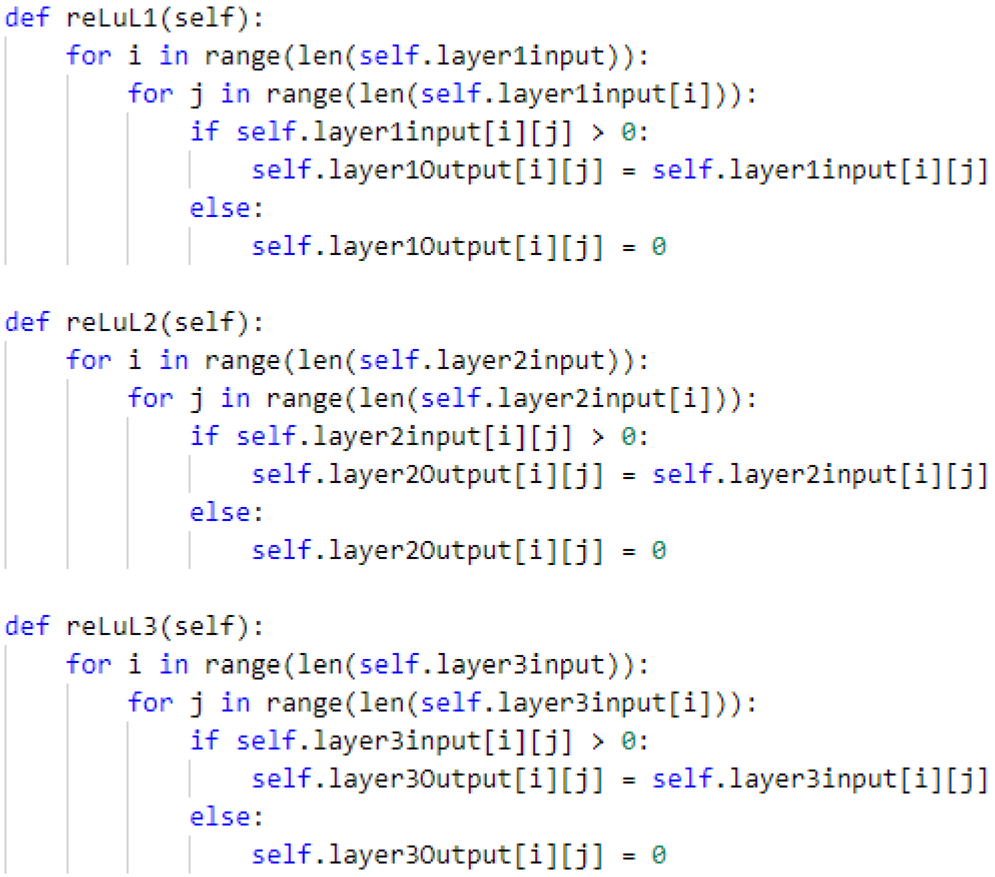
```
def feedForward(self):

    #Existing Dot Calcualation
    #self.layer1input = np.dot(self.input, self.obj_layer1.synapticWeights)
    self.layer1input = self.allRowsDotProduct(self.input,self.obj_layer1.synapticWeights, self.obj_layer1.bias)
    self.layer1Output = np.copy(self.layer1input)
    self.act_l1()

    if self.logoutput:
        toExcel = pd.DataFrame(self.layer1input)
        toExcel.to_excel(
            r'Logs\layer1input' + str(self.activation) + str(self.feedType) + str(self.iterationno)+'.xlsx')
        toExcel = pd.DataFrame(self.layer1Output)
        toExcel.to_excel(
            r'Logs\layer1output' + str(self.activation) + str(self.feedType) + str(self.iterationno)+'.xlsx')

    self.layer2input = self.allRowsDotProduct(self.layer1Output, self.obj_layer2.synapticWeights, self.obj_layer2.bias)
    self.layer2Output = np.copy(self.layer2input)
    self.act_l2()

    self.layer3Output = np.copy(self.layer3input)
    self.act_l3()
```



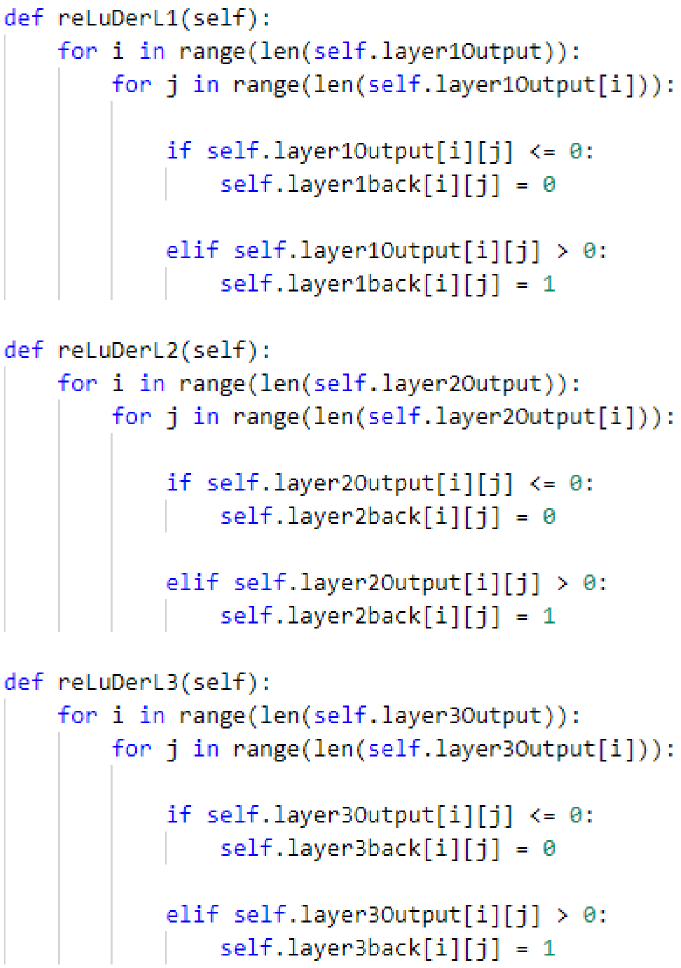
```python
def sigmoidL1(self):
    # define the sigmoid activation function

    for i in range(len(self.layer1input)):
        for j in range(len(self.layer1input[i])):

            if self.layer1input[i][j] >= 0:
                z = np.exp(-self.layer1input[i][j])
                self.layer1Output[i][j] = 1. / (1. + z)
            elif self.layer1input[i][j] < 0:
                z = np.exp(self.layer1input[i][j])
                self.layer1Output[i][j] = z / (1. + z)

def sigmoidL2(self):
    # define the sigmoid activation function

    for i in range(len(self.layer2input)):
        for j in range(len(self.layer2input[i])):

            if self.layer2input[i][j] >= 0:
                z = np.exp(-self.layer2input[i][j])
                self.layer2Output[i][j] = 1. / (1. + z)
            elif self.layer2input[i][j] < 0:
                z = np.exp(self.layer2input[i][j])
                self.layer2Output[i][j] = z / (1. + z)

def sigmoidL3(self):
    # define the sigmoid activation function

    for i in range(len(self.layer3input)):
        for j in range(len(self.layer3input[i])):

            if self.layer3input[i][j] >= 0:
                z = np.exp(-self.layer3input[i][j])
                self.layer3Output[i][j] = 1. / (1. + z)
            elif self.layer3input[i][j] < 0:
                z = np.exp(self.layer3input[i][j])
                self.layer3Output[i][j] = z / (1. + z)
```



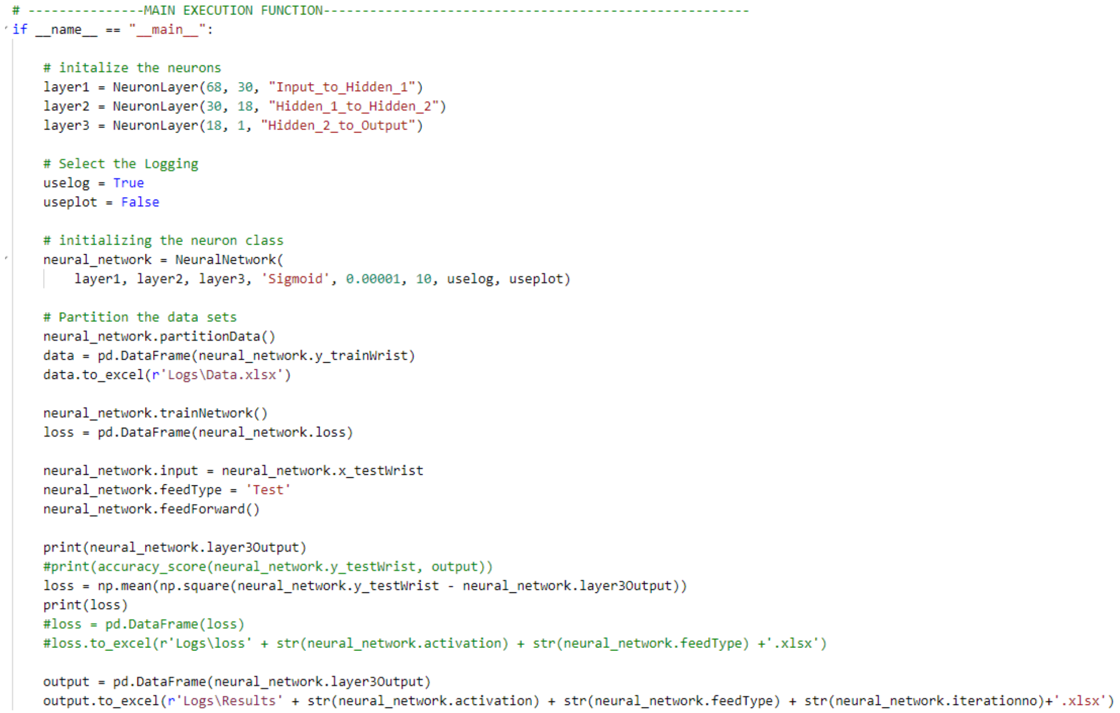
```python
def sigmoidDerivativeL1(self):
    # calculate the derivative of sigmoid
    self.layer1back = self.layer1Output * (1 - self.layer1Output)

def sigmoidDerivativeL2(self):
    # calculate the derivative of sigmoid
    self.layer2back = self.layer2Output * (1 - self.layer2Output)

def sigmoidDerivativeL3(self):
    # calculate the derivative of sigmoid
    self.layer3back = self.layer3Output * (1 - self.layer3Output)
```



```
def reLuL1(self):
    for i in range(len(self.layer1input)):
        for j in range(len(self.layer1input[i])):
            if self.layer1input[i][j] > 0:
                self.layer1Output[i][j] = self.layer1input[i][j]
            else:
                self.layer1Output[i][j] = 0

def reLuL2(self):
    for i in range(len(self.layer2input)):
        for j in range(len(self.layer2input[i])):
            if self.layer2input[i][j] > 0:
                self.layer2Output[i][j] = self.layer2input[i][j]
            else:
                self.layer2Output[i][j] = 0

def reLuL3(self):
    for i in range(len(self.layer3input)):
        for j in range(len(self.layer3input[i])):
            if self.layer3input[i][j] > 0:
                self.layer3Output[i][j] = self.layer3input[i][j]
            else:
                self.layer3Output[i][j] = 0
```

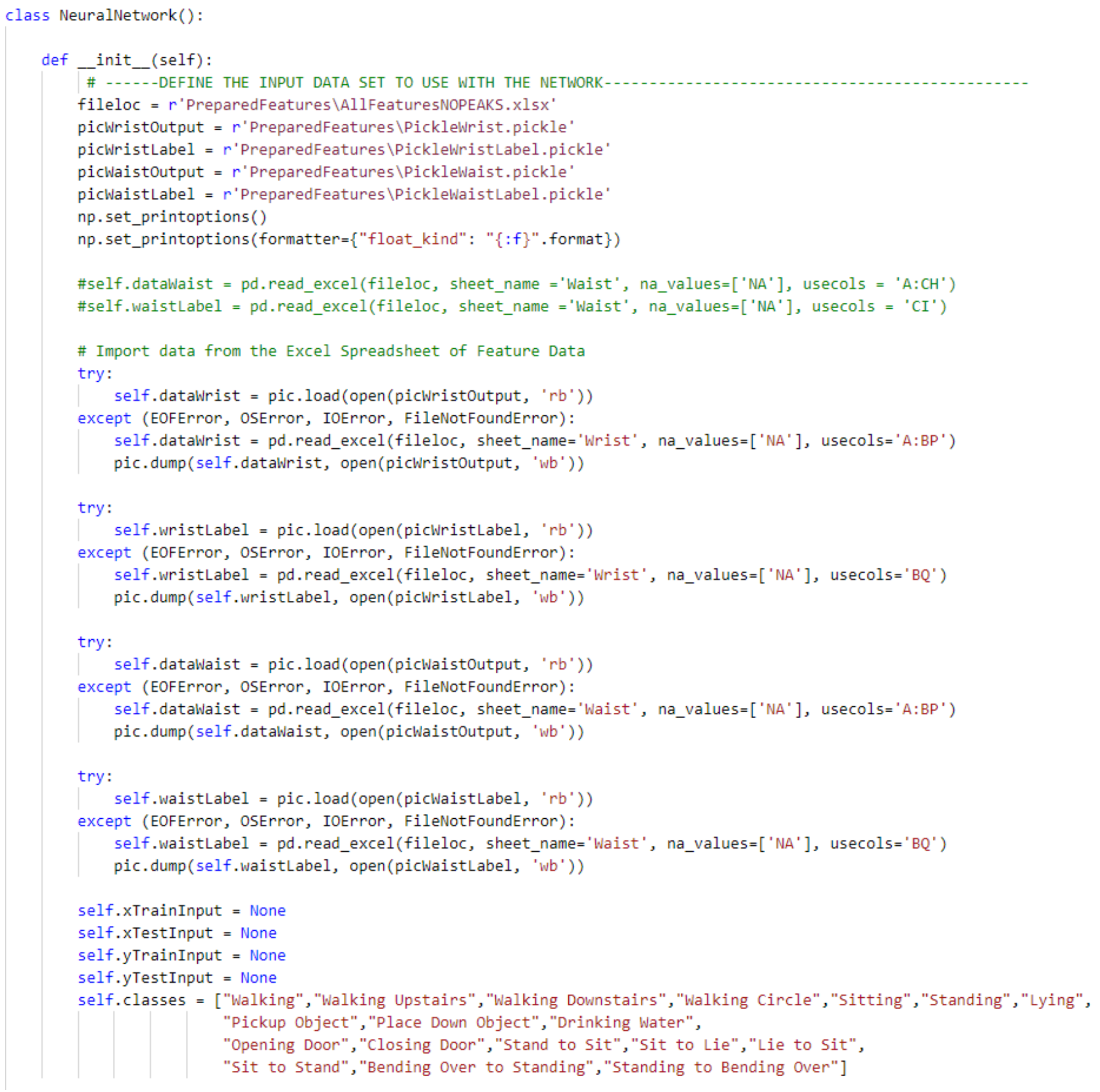


```
def reLuDerL1(self):
    for i in range(len(self.layer1Output)):
        for j in range(len(self.layer1Output[i])):

            if self.layer1Output[i][j] <= 0:
                self.layer1back[i][j] = 0

            elif self.layer1Output[i][j] > 0:
                self.layer1back[i][j] = 1

def reLuDerL2(self):
    for i in range(len(self.layer2Output)):
        for j in range(len(self.layer2Output[i])):

            if self.layer2Output[i][j] <= 0:
                self.layer2back[i][j] = 0

            elif self.layer2Output[i][j] > 0:
                self.layer2back[i][j] = 1

def reLuDerL3(self):
    for i in range(len(self.layer3Output)):
        for j in range(len(self.layer3Output[i])):

            if self.layer3Output[i][j] <= 0:
                self.layer3back[i][j] = 0

            elif self.layer3Output[i][j] > 0:
                self.layer3back[i][j] = 1
```

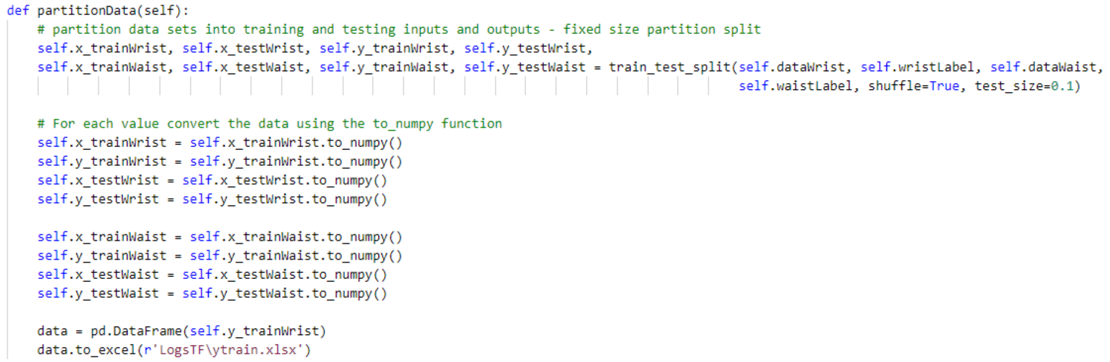


```
# ---------------MAIN EXECUTION FUNCTION-----------------------------------------------------
if __name__ == "__main__":

    # initalize the neurons
    layer1 = NeuronLayer(68, 30, "Input_to_Hidden_1")
    layer2 = NeuronLayer(30, 18, "Hidden_1_to_Hidden_2")
    layer3 = NeuronLayer(18, 1, "Hidden_2_to_Output")

    # Select the Logging
    uselog = True
    useplot = False

    # initializing the neuron class
    neural_network = NeuralNetwork(
        layer1, layer2, layer3, 'Sigmoid', 0.00001, 10, uselog, useplot)

    # Partition the data sets
    neural_network.partitionData()
    data = pd.DataFrame(neural_network.y_trainWrist)
    data.to_excel(r'Logs\Data.xlsx')

    neural_network.trainNetwork()
    loss = pd.DataFrame(neural_network.loss)

    neural_network.input = neural_network.x_testWrist
    neural_network.feedType = 'Test'
    neural_network.feedForward()

    print(neural_network.layer3Output)
    #print(accuracy_score(neural_network.y_testWrist, output))
    loss = np.mean(np.square(neural_network.y_testWrist - neural_network.layer3Output))
    print(loss)
    #loss = pd.DataFrame(loss)
    #loss.to_excel(r'Logs\loss' + str(neural_network.activation) + str(neural_network.feedType) +'.xlsx')

    output = pd.DataFrame(neural_network.layer3Output)
    output.to_excel(r'Logs\Results' + str(neural_network.activation) + str(neural_network.feedType) + str(neural_network.iterationno)+'.xlsx')
```

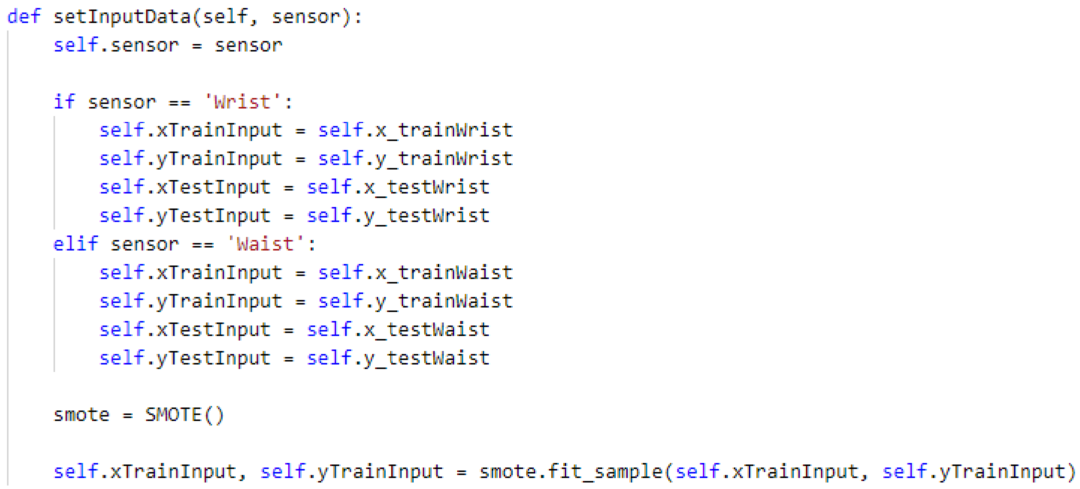
Appendix B – Framework Implementation for Convolutional Neural Network



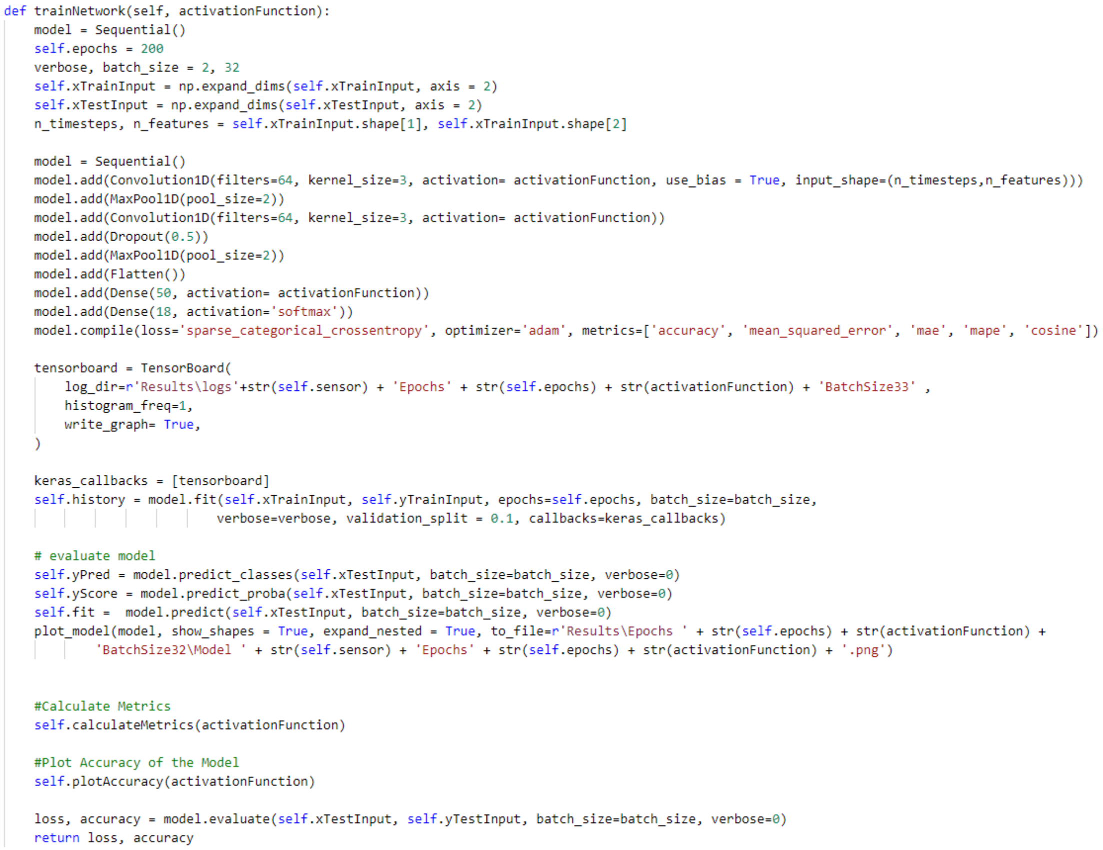
```
class NeuralNetwork():

    def __init__(self):
        # ------DEFINE THE INPUT DATA SET TO USE WITH THE NETWORK----------------------------------------------
        fileloc = r'PreparedFeatures\AllFeaturesNOPEAKS.xlsx'
        picWristOutput = r'PreparedFeatures\PickleWrist.pickle'
        picWristLabel = r'PreparedFeatures\PickleWristLabel.pickle'
        picWaistOutput = r'PreparedFeatures\PickleWaist.pickle'
        picWaistLabel = r'PreparedFeatures\PickleWaistLabel.pickle'
        np.set_printoptions()
        np.set_printoptions(formatter={"float_kind": "{:f}".format})

        #self.dataWaist = pd.read_excel(fileloc, sheet_name ='Waist', na_values=['NA'], usecols = 'A:CH')
        #self.waistLabel = pd.read_excel(fileloc, sheet_name ='Waist', na_values=['NA'], usecols = 'CI')

        # Import data from the Excel Spreadsheet of Feature Data
        try:
            self.dataWrist = pic.load(open(picWristOutput, 'rb'))
        except (EOFError, OSError, IOError, FileNotFoundError):
            self.dataWrist = pd.read_excel(fileloc, sheet_name='Wrist', na_values=['NA'], usecols='A:BP')
            pic.dump(self.dataWrist, open(picWristOutput, 'wb'))

        try:
            self.wristLabel = pic.load(open(picWristLabel, 'rb'))
        except (EOFError, OSError, IOError, FileNotFoundError):
            self.wristLabel = pd.read_excel(fileloc, sheet_name='Wrist', na_values=['NA'], usecols='BQ')
            pic.dump(self.wristLabel, open(picWristLabel, 'wb'))

        try:
            self.dataWaist = pic.load(open(picWaistOutput, 'rb'))
        except (EOFError, OSError, IOError, FileNotFoundError):
            self.dataWaist = pd.read_excel(fileloc, sheet_name='Waist', na_values=['NA'], usecols='A:BP')
            pic.dump(self.dataWaist, open(picWaistOutput, 'wb'))

        try:
            self.waistLabel = pic.load(open(picWaistLabel, 'rb'))
        except (EOFError, OSError, IOError, FileNotFoundError):
            self.waistLabel = pd.read_excel(fileloc, sheet_name='Waist', na_values=['NA'], usecols='BQ')
            pic.dump(self.waistLabel, open(picWaistLabel, 'wb'))

        self.xTrainInput = None
        self.xTestInput = None
        self.yTrainInput = None
        self.yTestInput = None
        self.classes = ["Walking","Walking Upstairs","Walking Downstairs","Walking Circle","Sitting","Standing","Lying",
                        "Pickup Object","Place Down Object","Drinking Water",
                        "Opening Door","Closing Door","Stand to Sit","Sit to Lie","Lie to Sit",
                        "Sit to Stand","Bending Over to Standing","Standing to Bending Over"]
```



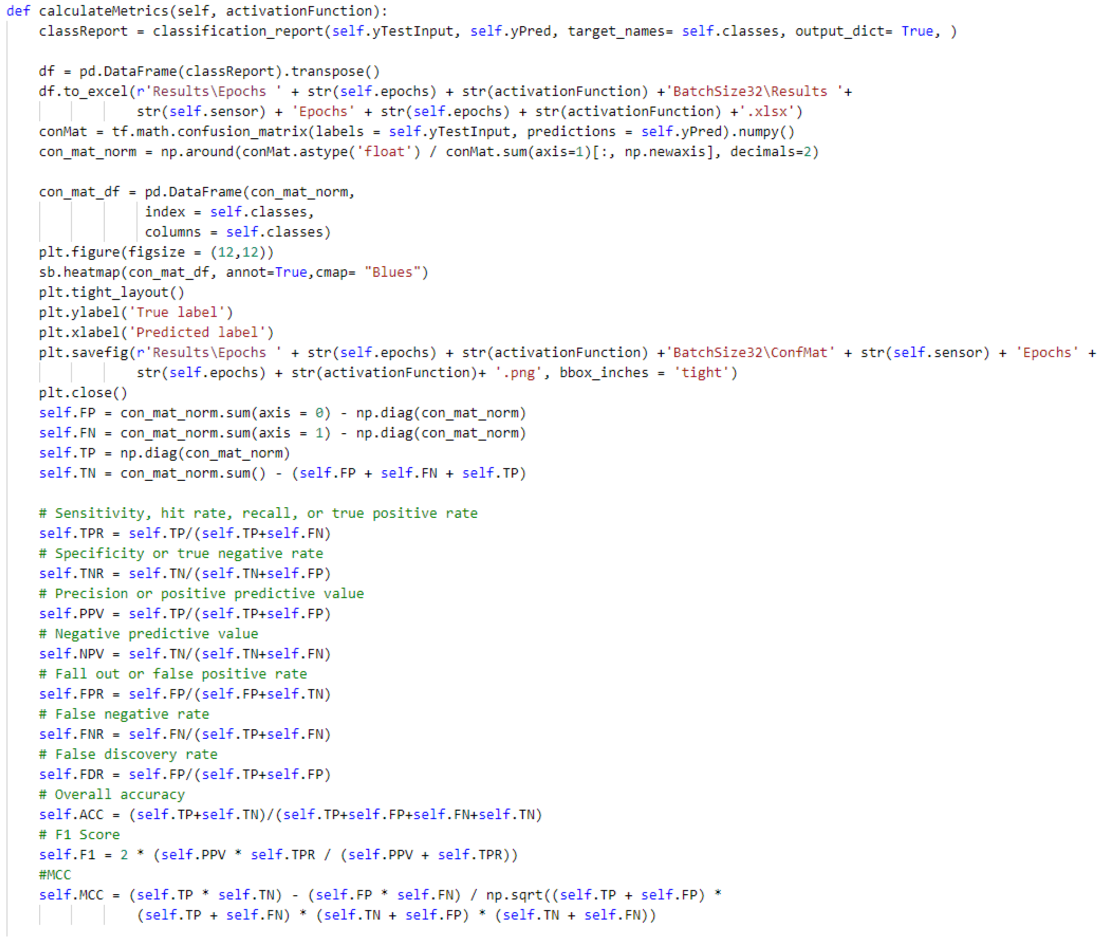
```
def partitionData(self):
    # partition data sets into training and testing inputs and outputs - fixed size partition split
    self.x_trainWrist, self.x_testWrist, self.y_trainWrist, self.y_testWrist,
    self.x_trainWaist, self.x_testWaist, self.y_trainWaist, self.y_testWaist = train_test_split(self.dataWrist, self.wristLabel, self.dataWaist,
                                                                                                self.waistLabel, shuffle=True, test_size=0.1)

    # For each value convert the data using the to_numpy function
    self.x_trainWrist = self.x_trainWrist.to_numpy()
    self.y_trainWrist = self.y_trainWrist.to_numpy()
    self.x_testWrist = self.x_testWrist.to_numpy()
    self.y_testWrist = self.y_testWrist.to_numpy()

    self.x_trainWaist = self.x_trainWaist.to_numpy()
    self.y_trainWaist = self.y_trainWaist.to_numpy()
    self.x_testWaist = self.x_testWaist.to_numpy()
    self.y_testWaist = self.y_testWaist.to_numpy()

    data = pd.DataFrame(self.y_trainWrist)
    data.to_excel(r'LogsTF\ytrain.xlsx')
```



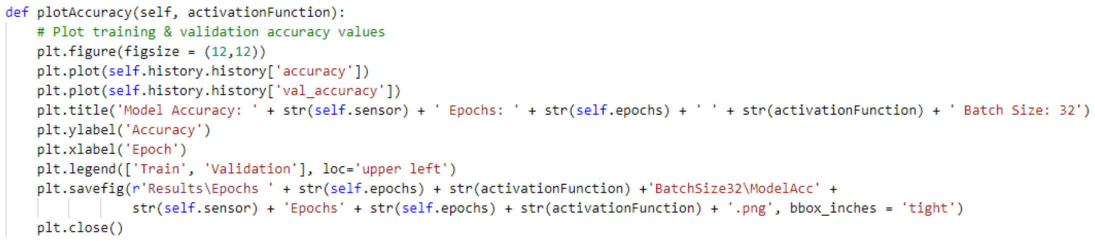
```
def setInputData(self, sensor):
    self.sensor = sensor

    if sensor == 'Wrist':
        self.xTrainInput = self.x_trainWrist
        self.yTrainInput = self.y_trainWrist
        self.xTestInput = self.x_testWrist
        self.yTestInput = self.y_testWrist
    elif sensor == 'Waist':
        self.xTrainInput = self.x_trainWaist
        self.yTrainInput = self.y_trainWaist
        self.xTestInput = self.x_testWaist
        self.yTestInput = self.y_testWaist

    smote = SMOTE()

    self.xTrainInput, self.yTrainInput = smote.fit_sample(self.xTrainInput, self.yTrainInput)
```



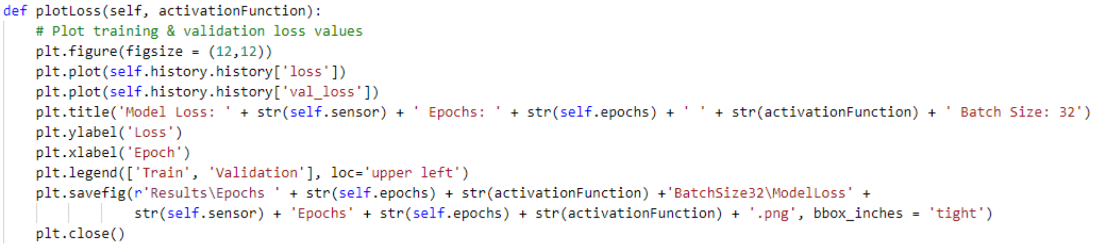
```
def trainNetwork(self, activationFunction):
    model = Sequential()
    self.epochs = 200
    verbose, batch_size = 2, 32
    self.xTrainInput = np.expand_dims(self.xTrainInput, axis = 2)
    self.xTestInput = np.expand_dims(self.xTestInput, axis = 2)
    n_timesteps, n_features = self.xTrainInput.shape[1], self.xTrainInput.shape[2]

    model = Sequential()
    model.add(Convolution1D(filters=64, kernel_size=3, activation= activationFunction, use_bias = True, input_shape=(n_timesteps,n_features)))
    model.add(MaxPool1D(pool_size=2))
    model.add(Convolution1D(filters=64, kernel_size=3, activation= activationFunction))
    model.add(Dropout(0.5))
    model.add(MaxPool1D(pool_size=2))
    model.add(Flatten())
    model.add(Dense(50, activation= activationFunction))
    model.add(Dense(18, activation='softmax'))
    model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy', 'mean_squared_error', 'mae', 'mape', 'cosine'])

    tensorboard = TensorBoard(
        log_dir=r'Results\logs'+str(self.sensor) + 'Epochs' + str(self.epochs) + str(activationFunction) + 'BatchSize33' ,
        histogram_freq=1,
        write_graph= True,
    )

    keras_callbacks = [tensorboard]
    self.history = model.fit(self.xTrainInput, self.yTrainInput, epochs=self.epochs, batch_size=batch_size,
                            verbose=verbose, validation_split = 0.1, callbacks=keras_callbacks)

    # evaluate model
    self.yPred = model.predict_classes(self.xTestInput, batch_size=batch_size, verbose=0)
    self.yScore = model.predict_proba(self.xTestInput, batch_size=batch_size, verbose=0)
    self.fit =  model.predict(self.xTestInput, batch_size=batch_size, verbose=0)
    plot_model(model, show_shapes = True, expand_nested = True, to_file=r'Results\Epochs ' + str(self.epochs) + str(activationFunction) +
            'BatchSize32\Model ' + str(self.sensor) + 'Epochs' + str(self.epochs) + str(activationFunction) + '.png')


    #Calculate Metrics
    self.calculateMetrics(activationFunction)

    #Plot Accuracy of the Model
    self.plotAccuracy(activationFunction)

    loss, accuracy = model.evaluate(self.xTestInput, self.yTestInput, batch_size=batch_size, verbose=0)
    return loss, accuracy
```



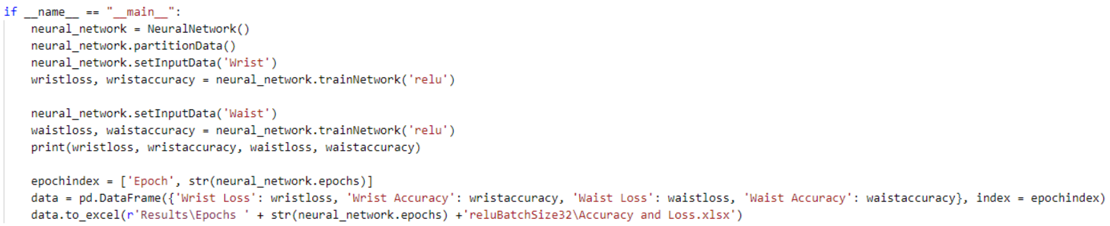
```python
def calculateMetrics(self, activationFunction):
    classReport = classification_report(self.yTestInput, self.yPred, target_names= self.classes, output_dict= True, )

    df = pd.DataFrame(classReport).transpose()
    df.to_excel(r'Results\Epochs ' + str(self.epochs) + str(activationFunction) +'BatchSize32\Results '+
                str(self.sensor) + 'Epochs' + str(self.epochs) + str(activationFunction) +'.xlsx')
    conMat = tf.math.confusion_matrix(labels = self.yTestInput, predictions = self.yPred).numpy()
    con_mat_norm = np.around(conMat.astype('float') / conMat.sum(axis=1)[:, np.newaxis], decimals=2)

    con_mat_df = pd.DataFrame(con_mat_norm,
                 index = self.classes,
                 columns = self.classes)
    plt.figure(figsize = (12,12))
    sb.heatmap(con_mat_df, annot=True,cmap= "Blues")
    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
    plt.savefig(r'Results\Epochs ' + str(self.epochs) + str(activationFunction) +'BatchSize32\ConfMat' + str(self.sensor) + 'Epochs' +
                str(self.epochs) + str(activationFunction)+ '.png', bbox_inches = 'tight')
    plt.close()
    self.FP = con_mat_norm.sum(axis = 0) - np.diag(con_mat_norm)
    self.FN = con_mat_norm.sum(axis = 1) - np.diag(con_mat_norm)
    self.TP = np.diag(con_mat_norm)
    self.TN = con_mat_norm.sum() - (self.FP + self.FN + self.TP)

    # Sensitivity, hit rate, recall, or true positive rate
    self.TPR = self.TP/(self.TP+self.FN)
    # Specificity or true negative rate
    self.TNR = self.TN/(self.TN+self.FP)
    # Precision or positive predictive value
    self.PPV = self.TP/(self.TP+self.FP)
    # Negative predictive value
    self.NPV = self.TN/(self.TN+self.FN)
    # Fall out or false positive rate
    self.FPR = self.FP/(self.FP+self.TN)
    # False negative rate
    self.FNR = self.FN/(self.TP+self.FN)
    # False discovery rate
    self.FDR = self.FP/(self.TP+self.FP)
    # Overall accuracy
    self.ACC = (self.TP+self.TN)/(self.TP+self.FP+self.FN+self.TN)
    # F1 Score
    self.F1 = 2 * (self.PPV * self.TPR / (self.PPV + self.TPR))
    #MCC
    self.MCC = (self.TP * self.TN) - (self.FP * self.FN) / np.sqrt((self.TP + self.FP) *
                (self.TP + self.FN) * (self.TN + self.FP) * (self.TN + self.FN))
```



```python
def plotAccuracy(self, activationFunction):
    # Plot training & validation accuracy values
    plt.figure(figsize = (12,12))
    plt.plot(self.history.history['accuracy'])
    plt.plot(self.history.history['val_accuracy'])
    plt.title('Model Accuracy: ' + str(self.sensor) + ' Epochs: ' + str(self.epochs) + ' ' + str(activationFunction) + ' Batch Size: 32')
    plt.ylabel('Accuracy')
    plt.xlabel('Epoch')
    plt.legend(['Train', 'Validation'], loc='upper left')
    plt.savefig(r'Results\Epochs ' + str(self.epochs) + str(activationFunction) +'BatchSize32\ModelAcc' +
                str(self.sensor) + 'Epochs' + str(self.epochs) + str(activationFunction) + '.png', bbox_inches = 'tight')
    plt.close()
```



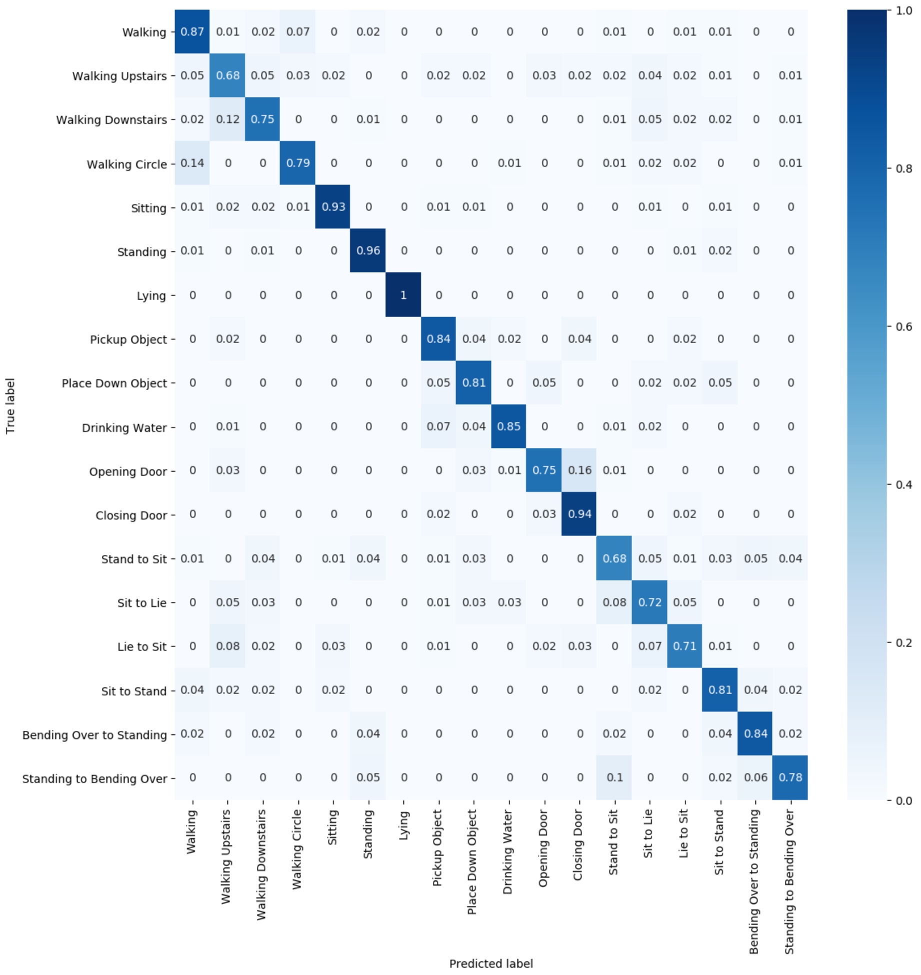
```python
def plotLoss(self, activationFunction):
    # Plot training & validation loss values
    plt.figure(figsize = (12,12))
    plt.plot(self.history.history['loss'])
    plt.plot(self.history.history['val_loss'])
    plt.title('Model Loss: ' + str(self.sensor) + ' Epochs: ' + str(self.epochs) + ' ' + str(activationFunction) + ' Batch Size: 32')
    plt.ylabel('Loss')
    plt.xlabel('Epoch')
    plt.legend(['Train', 'Validation'], loc='upper left')
    plt.savefig(r'Results\Epochs ' + str(self.epochs) + str(activationFunction) +'BatchSize32\ModelLoss' +
                str(self.sensor) + 'Epochs' + str(self.epochs) + str(activationFunction) + '.png', bbox_inches = 'tight')
    plt.close()
```



```python
if __name__ == "__main__":
    neural_network = NeuralNetwork()
    neural_network.partitionData()
    neural_network.setInputData('Wrist')
    wristloss, wristaccuracy = neural_network.trainNetwork('relu')

    neural_network.setInputData('Waist')
    waistloss, waistaccuracy = neural_network.trainNetwork('relu')
    print(wristloss, wristaccuracy, waistloss, waistaccuracy)

    epochindex = ['Epoch', str(neural_network.epochs)]
    data = pd.DataFrame({'Wrist Loss': wristloss, 'Wrist Accuracy': wristaccuracy, 'Waist Loss': waistloss, 'Waist Accuracy': waistaccuracy}, index = epochindex)
    data.to_excel(r'Results\Epochs ' + str(neural_network.epochs) +'reluBatchSize32\Accuracy and Loss.xlsx')
```

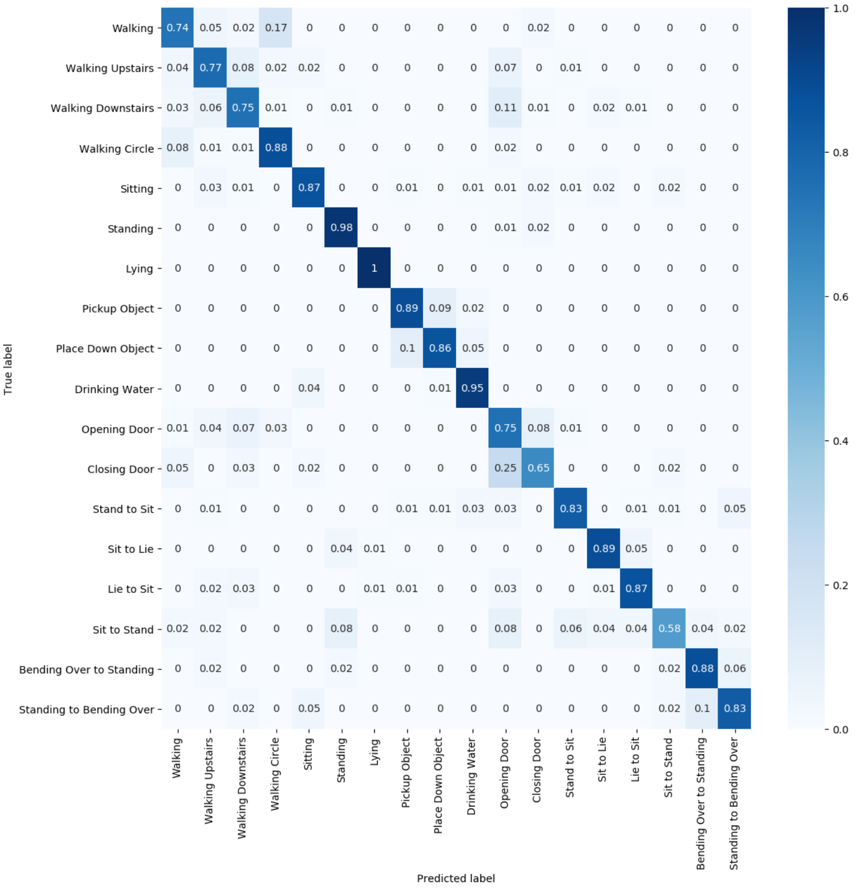
Appendix C – Additional Graphs: ReLu 100 Epochs

Confusion matrix for the wrist data set



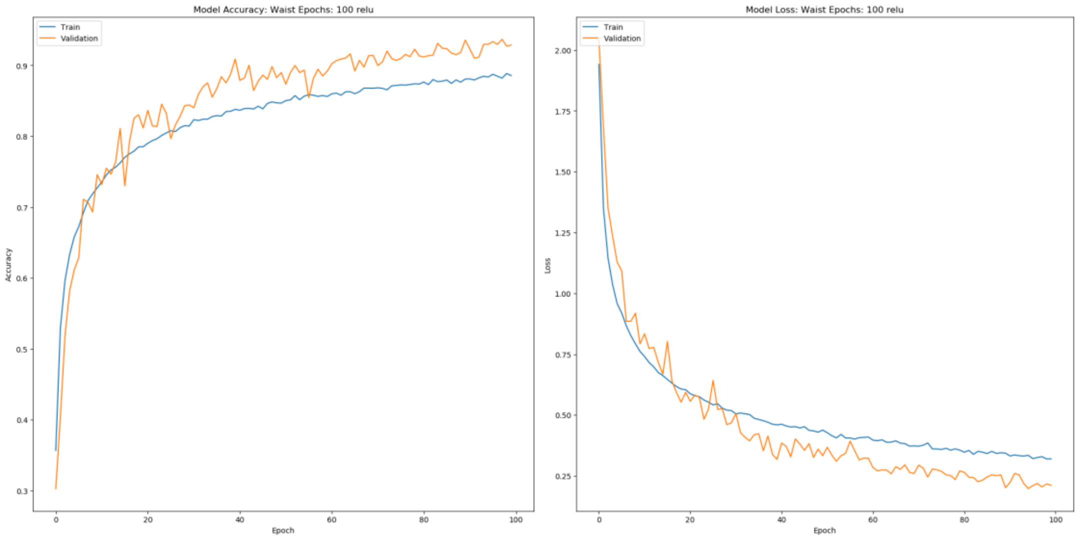
	Walking	Walking Upstairs	Walking Downstairs	Walking Circle	Sitting	Standing	Lying	Pickup Object	Place Down Object	Drinking Water	Opening Door	Closing Door	Stand to Sit	Sit to Lie	Lie to Sit	Sit to Stand	Bending Over to Standing	Standing to Bending Over
Walking	0.87	0.01	0.02	0.07	0	0.02	0	0	0	0	0	0	0.01	0	0.01	0.01	0	0
Walking Upstairs	0.05	0.68	0.05	0.03	0.02	0	0	0.02	0.02	0	0.03	0.02	0.02	0.04	0.02	0.01	0	0.01
Walking Downstairs	0.02	0.12	0.75	0	0	0.01	0	0	0	0	0	0	0.01	0.05	0.02	0.02	0	0.01
Walking Circle	0.14	0	0	0.79	0	0	0	0	0	0.01	0	0	0.01	0.02	0.02	0	0	0.01
Sitting	0.01	0.02	0.02	0.01	0.93	0	0	0.01	0.01	0	0	0	0	0.01	0	0.01	0	0
Standing	0.01	0	0.01	0	0	0.96	0	0	0	0	0	0	0	0	0.01	0.02	0	0
Lying	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Pickup Object	0	0.02	0	0	0	0	0	0.84	0.04	0.02	0	0.04	0	0	0.02	0	0	0
Place Down Object	0	0	0	0	0	0	0	0.05	0.81	0	0.05	0	0	0.02	0.02	0.05	0	0
Drinking Water	0	0.01	0	0	0	0	0	0.07	0.04	0.85	0	0	0.01	0.02	0	0	0	0
Opening Door	0	0.03	0	0	0	0	0	0	0.03	0.01	0.75	0.16	0.01	0	0	0	0	0
Closing Door	0	0	0	0	0	0	0	0.02	0	0	0.03	0.94	0	0	0.02	0	0	0
Stand to Sit	0.01	0	0.04	0	0.01	0.04	0	0.01	0.03	0	0	0	0.68	0.05	0.01	0.03	0.05	0.04
Sit to Lie	0	0.05	0.03	0	0	0	0	0.01	0.03	0.03	0	0	0.08	0.72	0.05	0	0	0
Lie to Sit	0	0.08	0.02	0	0.03	0	0	0.01	0	0	0.02	0.03	0	0.07	0.71	0.01	0	0
Sit to Stand	0.04	0.02	0.02	0	0.02	0	0	0	0	0	0	0	0	0.02	0	0.81	0.04	0.02
Bending Over to Standing	0.02	0	0.02	0	0	0.04	0	0	0	0	0	0	0.02	0	0	0.04	0.84	0.02
Standing to Bending Over	0	0	0	0	0	0.05	0	0	0	0	0	0	0.1	0	0	0.02	0.06	0.78

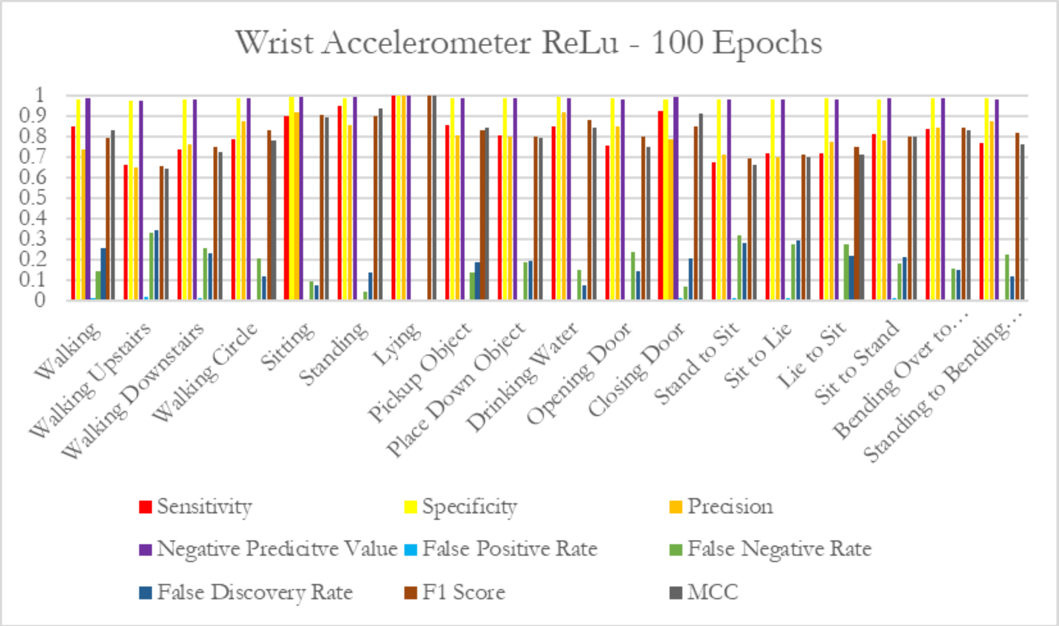
Confusion matrix for the waist data set

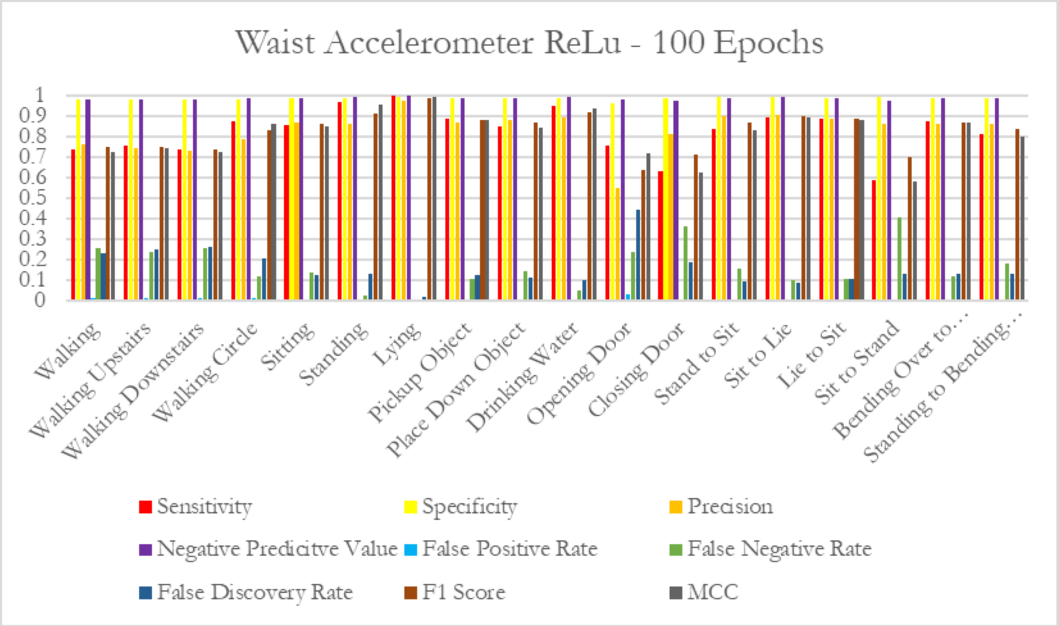


	Walking	Walking Upstairs	Walking Downstairs	Walking Circle	Sitting	Standing	Lying	Pickup Object	Place Down Object	Drinking Water	Opening Door	Closing Door	Stand to Sit	Sit to Lie	Lie to Sit	Sit to Stand	Bending Over to Standing	Standing to Bending Over
Walking	0.74	0.05	0.02	0.17	0	0	0	0	0	0	0	0.02	0	0	0	0	0	0
Walking Upstairs	0.04	0.77	0.08	0.02	0.02	0	0	0	0	0	0.07	0	0.01	0	0	0	0	0
Walking Downstairs	0.03	0.06	0.75	0.01	0	0.01	0	0	0	0	0.11	0.01	0	0.02	0.01	0	0	0
Walking Circle	0.08	0.01	0.01	0.88	0	0	0	0	0	0	0.02	0	0	0	0	0	0	0
Sitting	0	0.03	0.01	0	0.87	0	0	0.01	0	0.01	0.01	0.02	0.01	0.02	0	0.02	0	0
Standing	0	0	0	0	0	0.98	0	0	0	0	0.01	0.02	0	0	0	0	0	0
Lying	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Pickup Object	0	0	0	0	0	0	0	0.89	0.09	0.02	0	0	0	0	0	0	0	0
Place Down Object	0	0	0	0	0	0	0	0.1	0.86	0.05	0	0	0	0	0	0	0	0
Drinking Water	0	0	0	0	0.04	0	0	0	0.01	0.95	0	0	0	0	0	0	0	0
Opening Door	0.01	0.04	0.07	0.03	0	0	0	0	0	0	0.75	0.08	0.01	0	0	0	0	0
Closing Door	0.05	0	0.03	0	0.02	0	0	0	0	0	0.25	0.65	0	0	0	0.02	0	0
Stand to Sit	0	0.01	0	0	0	0	0	0.01	0.01	0.03	0.03	0	0.83	0	0.01	0.01	0	0.05
Sit to Lie	0	0	0	0	0	0.04	0.01	0	0	0	0	0	0	0.89	0.05	0	0	0
Lie to Sit	0	0.02	0.03	0	0	0	0.01	0.01	0	0	0.03	0	0	0.01	0.87	0	0	0
Sit to Stand	0.02	0.02	0	0	0	0.08	0	0	0	0	0.08	0	0.06	0.04	0.04	0.58	0.04	0.02
Bending Over to Standing	0	0.02	0	0	0	0.02	0	0	0	0	0	0	0	0	0	0.02	0.88	0.06
Standing to Bending Over	0	0	0.02	0	0.05	0	0	0	0	0	0	0	0	0	0	0.02	0.1	0.83

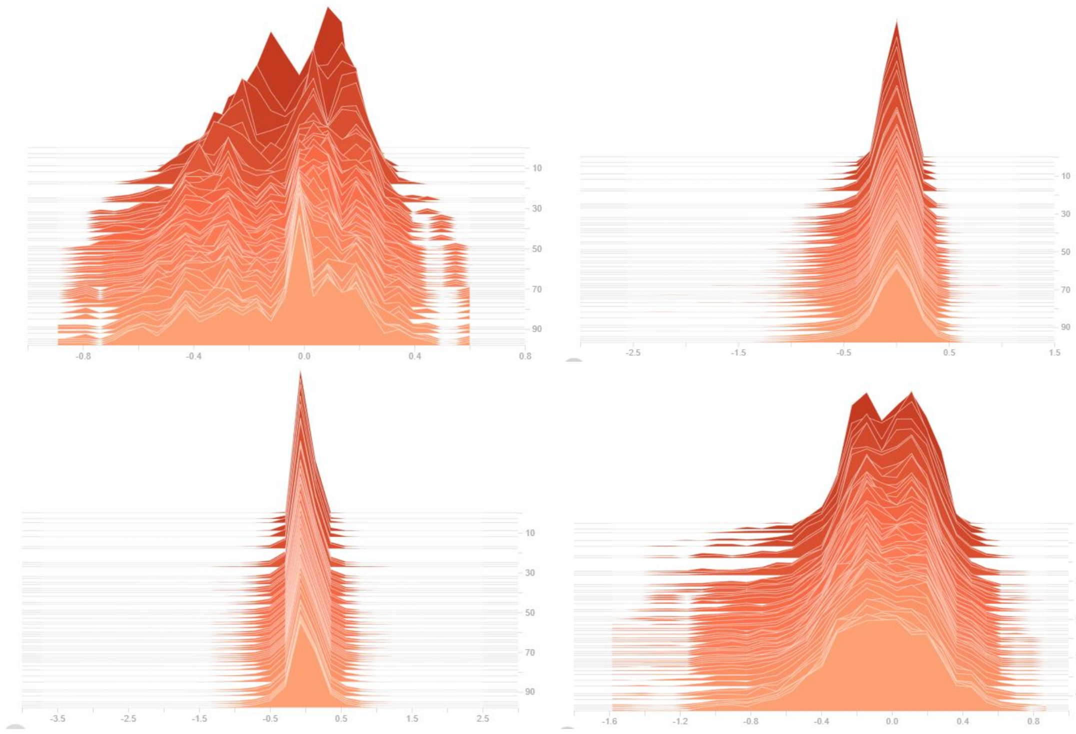
Waist Accuracy and Loss



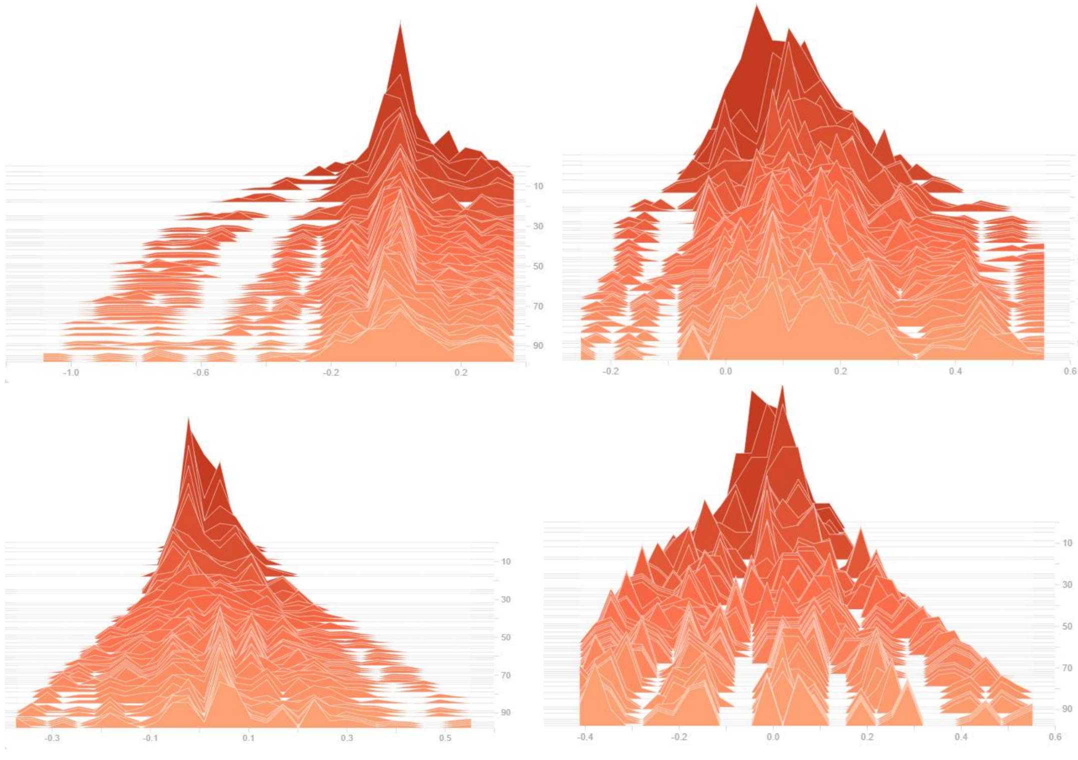

Wrist Accelerometer ReLu - 100 Epochs
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Walking
Walking Upstairs
Walking Downstairs
Walking Circle
Sitting
Standing
Lying
Pickup Object
Place Down Object
Drinking Water
Opening Door
Closing Door
Stand to Sit
Sit to Lie
Lie to Sit
Sit to Stand
Bending Over to…
Standing to Bending…
Sensitivity
Specificity
Precision
Negative Predicitve Value
False Positive Rate
False Negative Rate
False Discovery Rate
F1 Score
MCC


Waist Accelerometer ReLu - 100 Epochs
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Walking
Walking Upstairs
Walking Downstairs
Walking Circle
Sitting
Standing
Lying
Pickup Object
Place Down Object
Drinking Water
Opening Door
Closing Door
Stand to Sit
Sit to Lie
Lie to Sit
Sit to Stand
Bending Over to…
Standing to Bending…
Sensitivity
Specificity
Precision
Negative Predicitve Value
False Positive Rate
False Negative Rate
False Discovery Rate
F1 Score
MCC

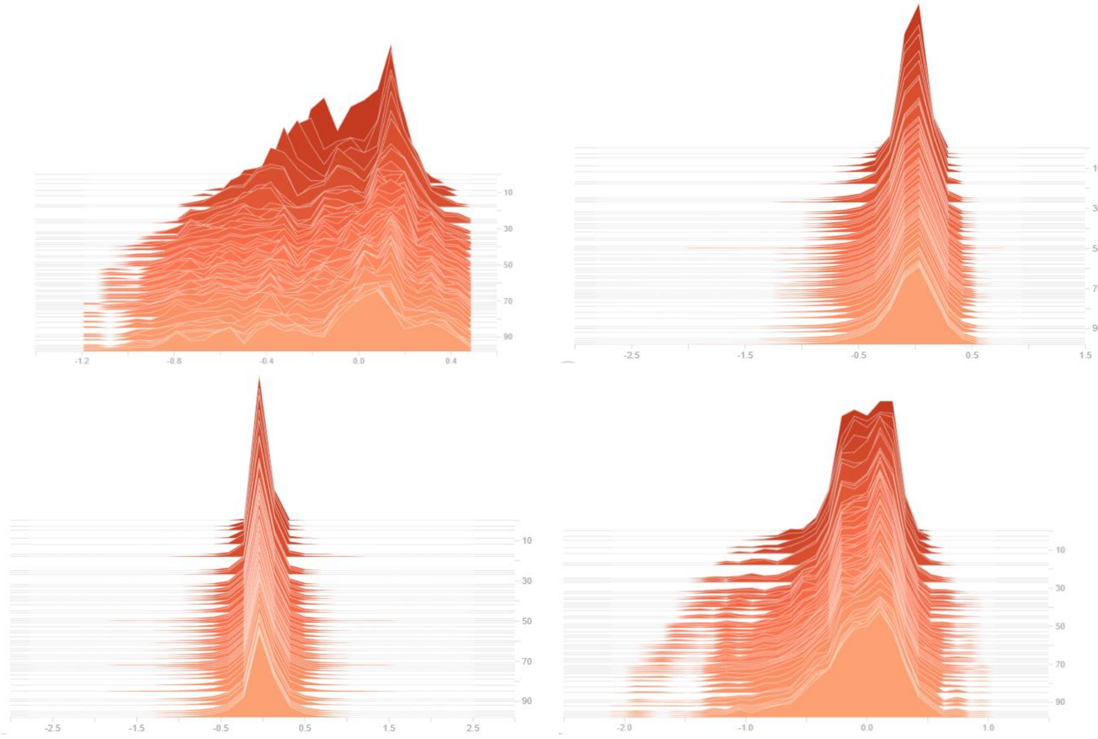
Wrist data set Histogram of kernel weights for the four layers of the network using ReLu at 100 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



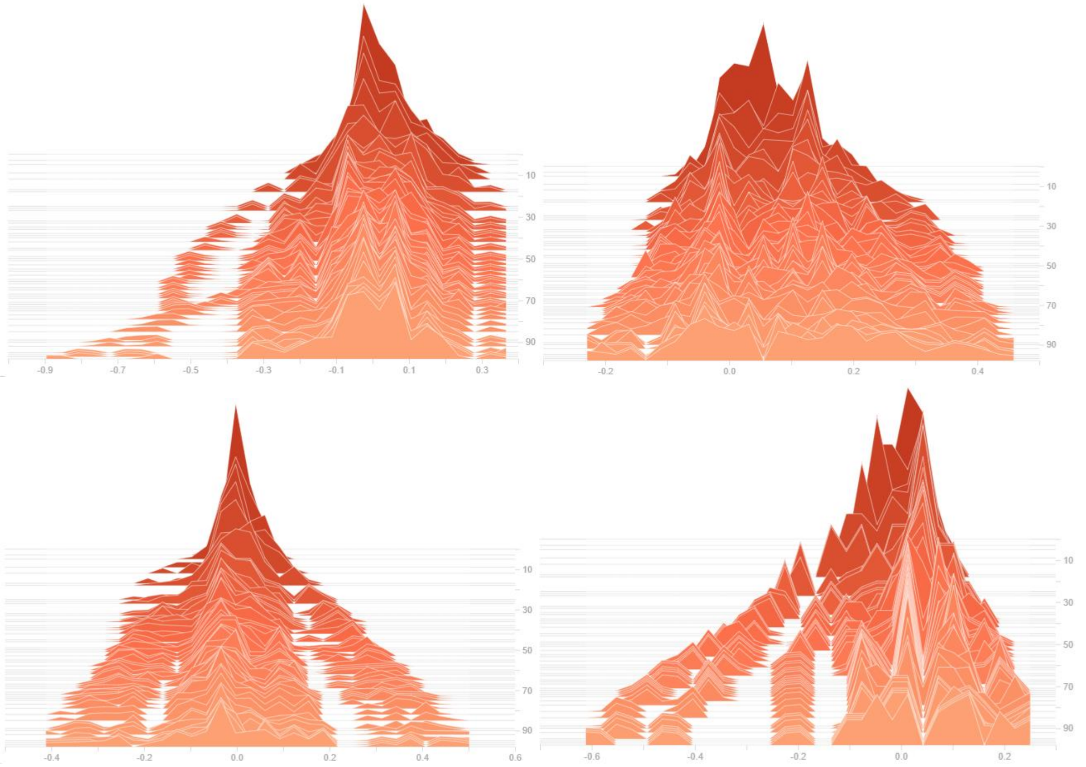
Wrist data set Histogram of bias weights for the four layers of the network using ReLu at 100 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



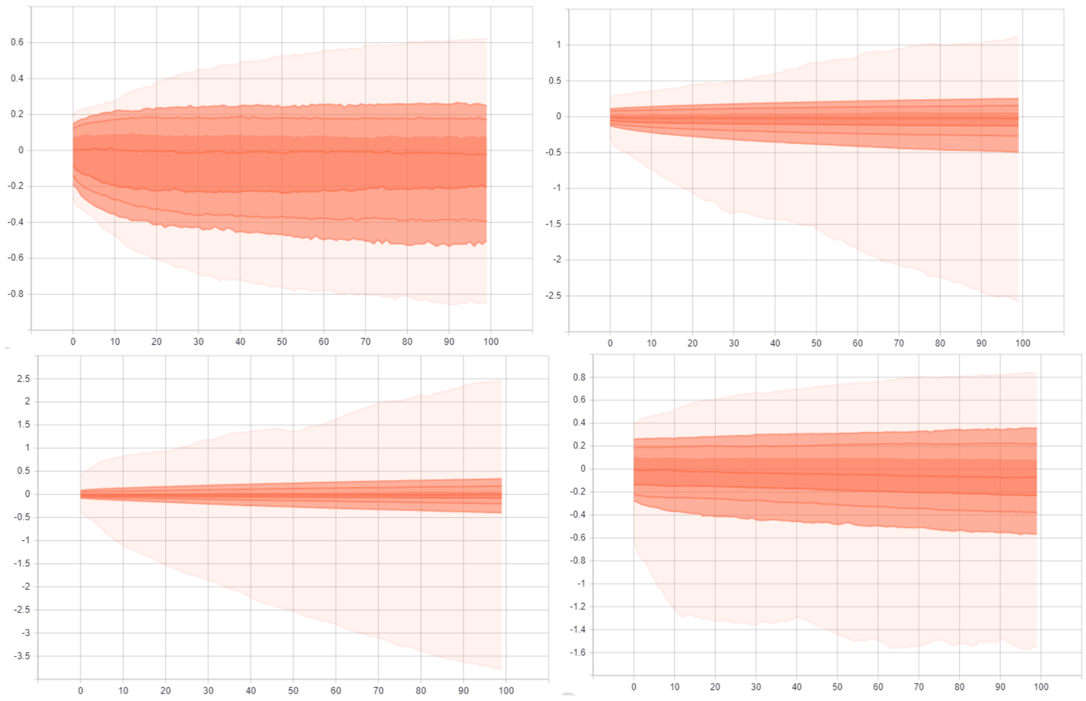
Waist data set Histogram of kernel weights for the four layers of the network using ReLu at 100 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



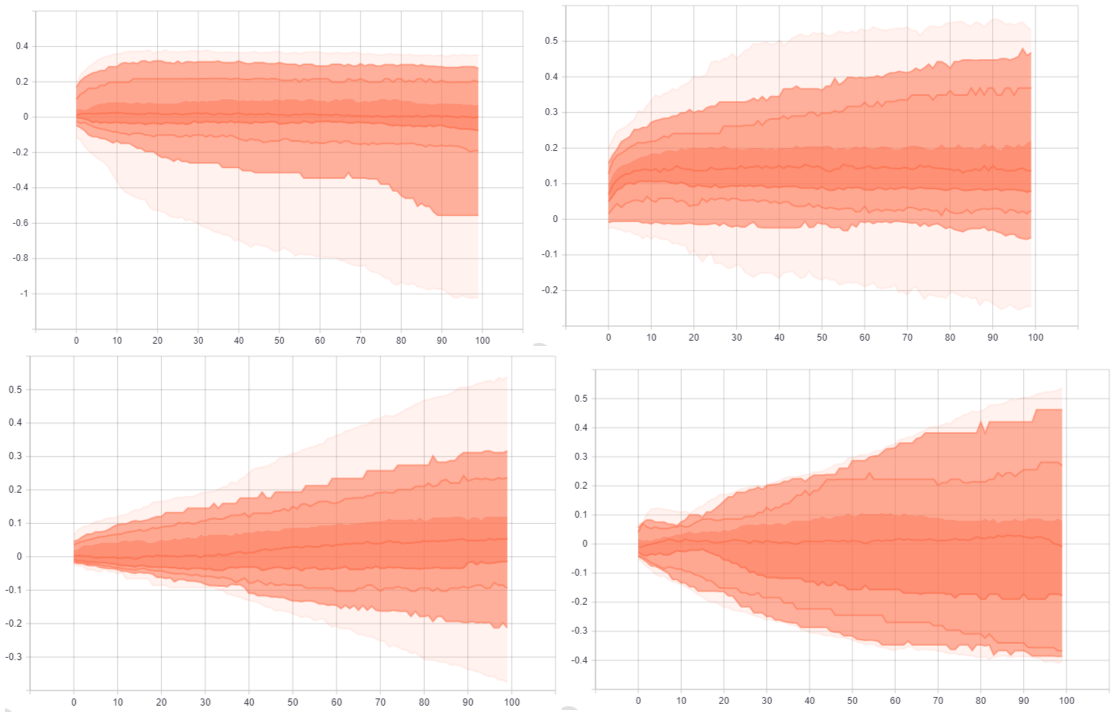
Waist data set Histogram of bias weights for the four layers of the network using ReLu at 100 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



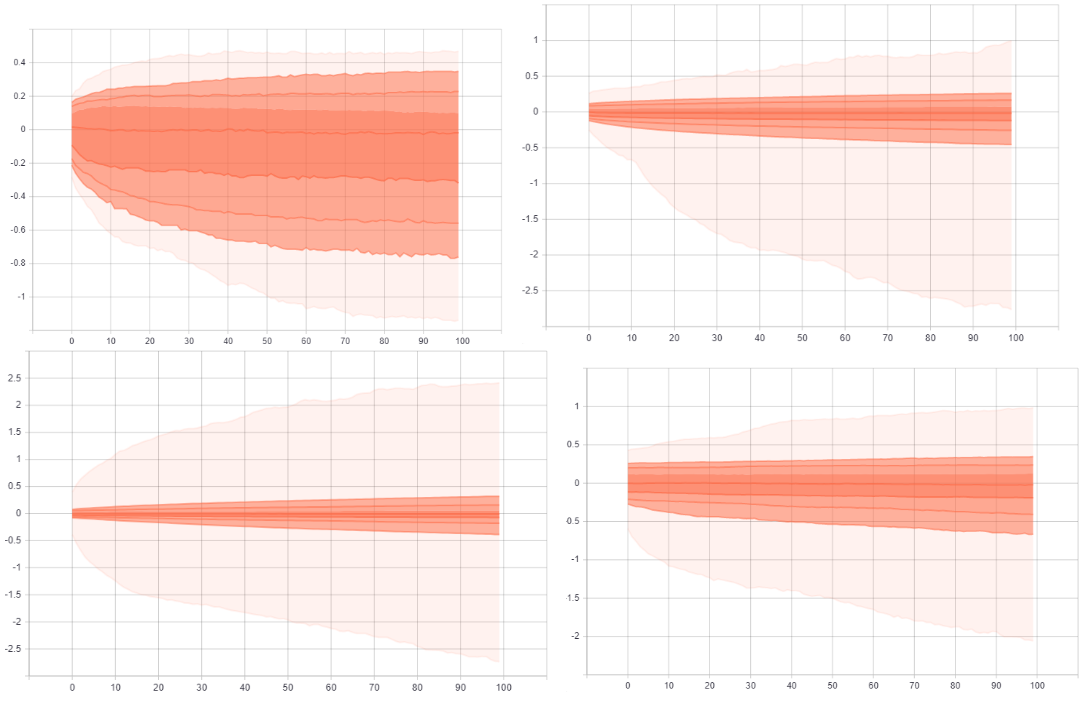
Wrist data set Distribution of kernel weights for the four layers of the network using ReLu at 100 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



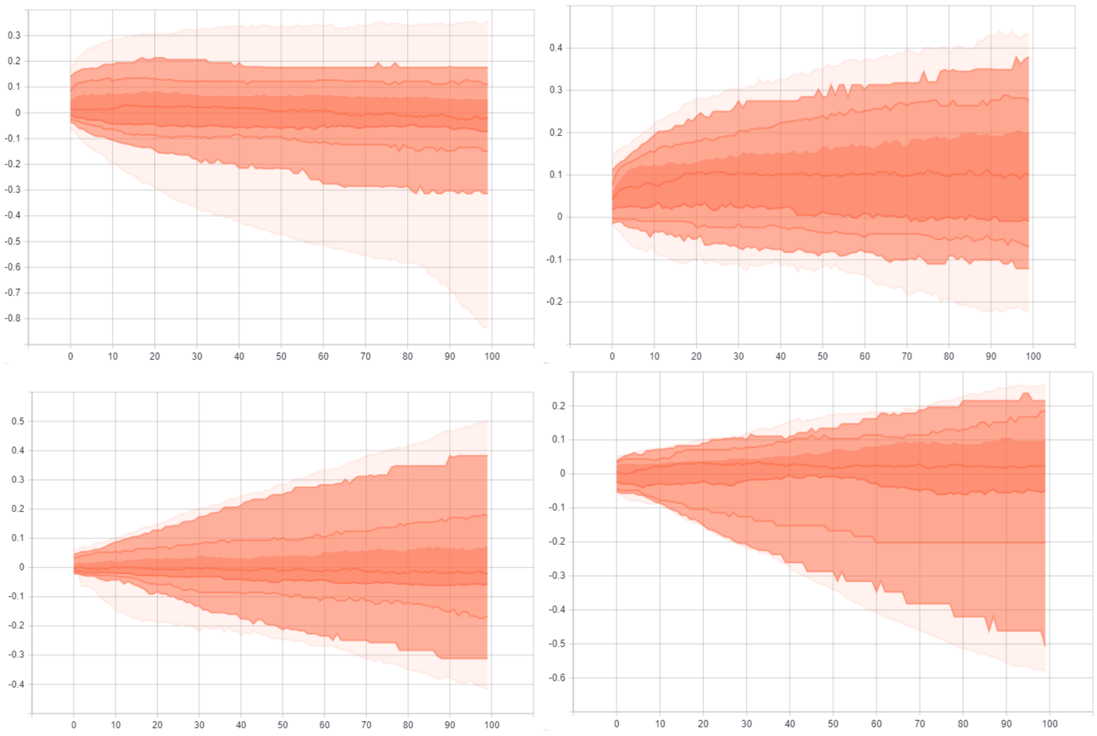
Wrist data set Distribution of bias weights for the four layers of the network using ReLu at 100 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



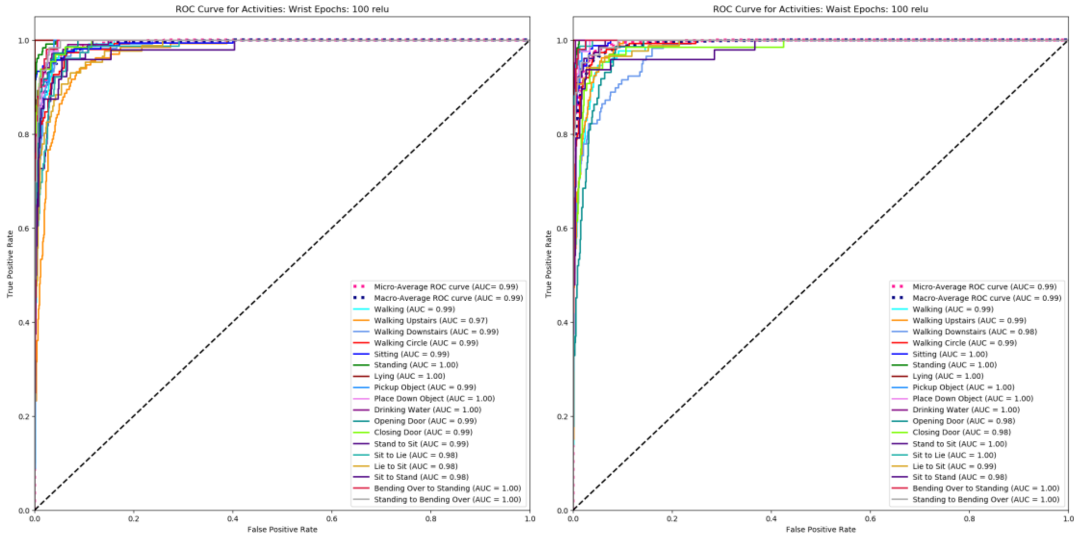
Waist data set Distribution of kernel weights for the four layers of the network using ReLu at 100 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



Waist data set Distribution of bias weights for the four layers of the network using ReLu at 100 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.

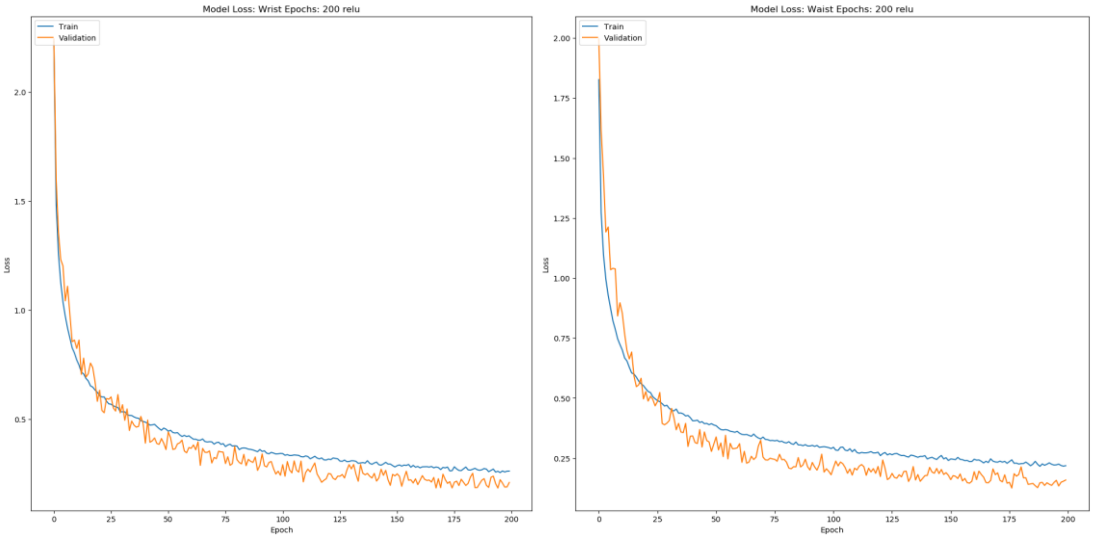


ROC Curve for Wrist and Waist

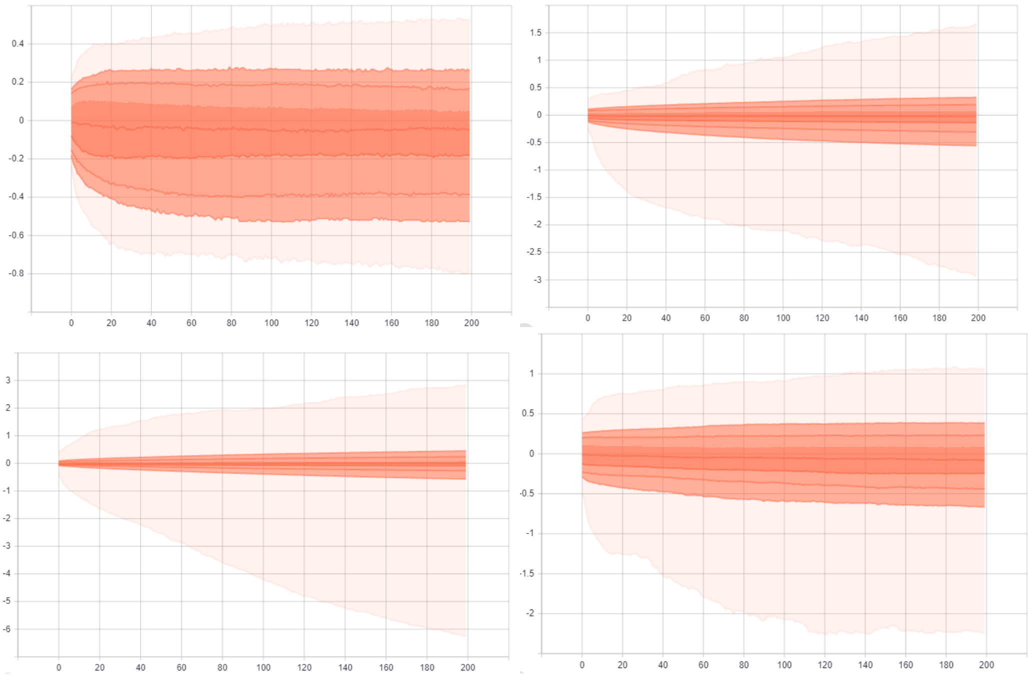


Appendix D – Additional Graphs: ReLu 200 Epochs

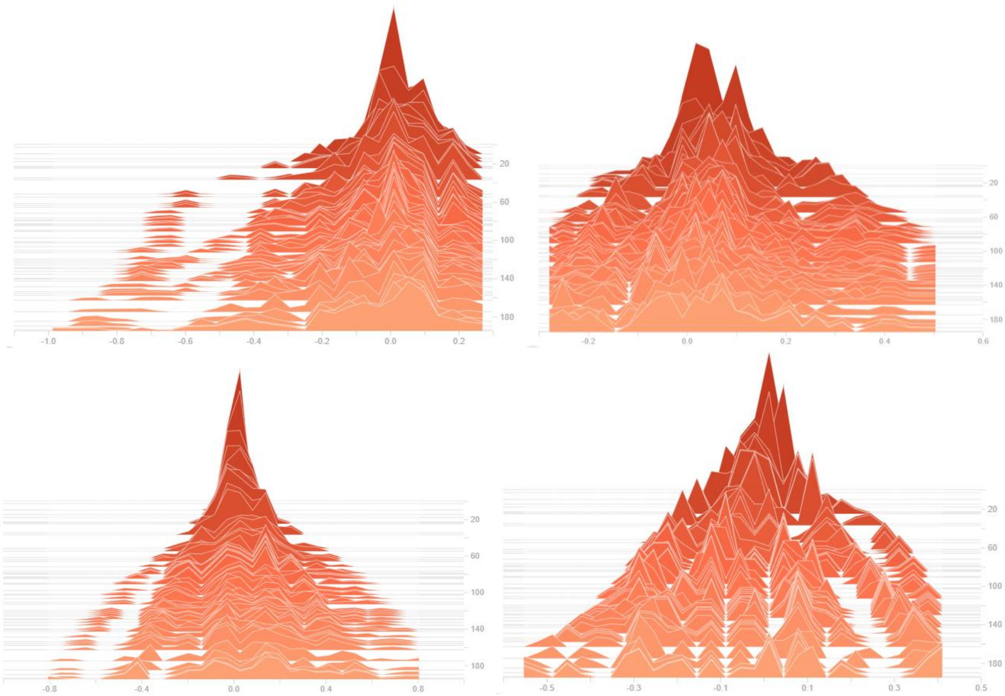
Loss for Wrist and Waist



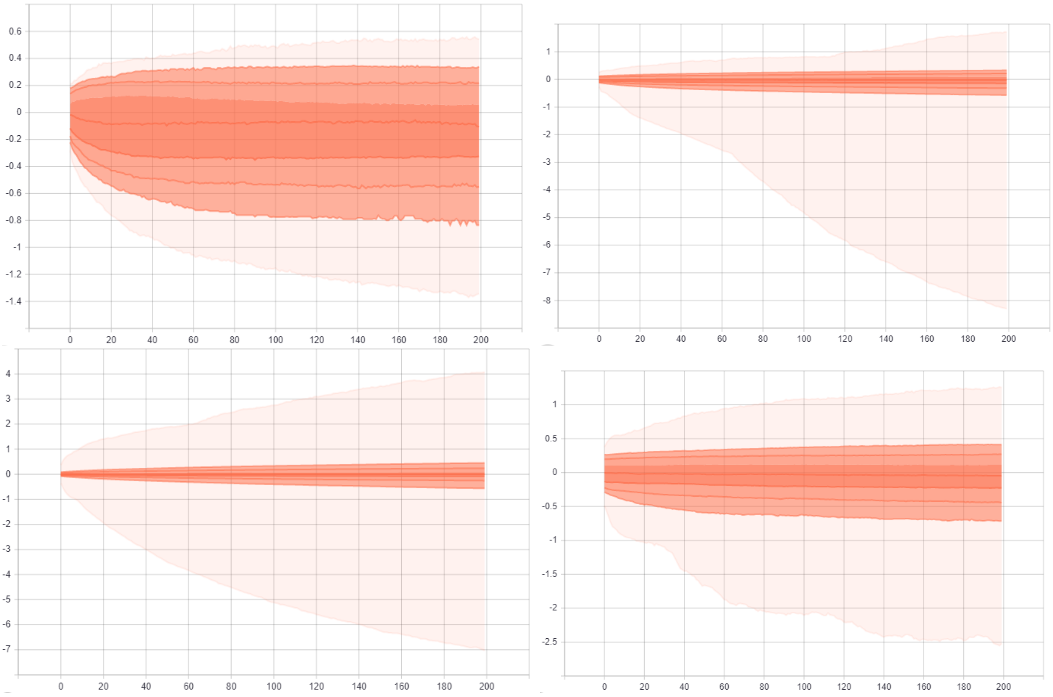
Wrist data set Distribution of kernel weights for the four layers of the network using ReLu at 200 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



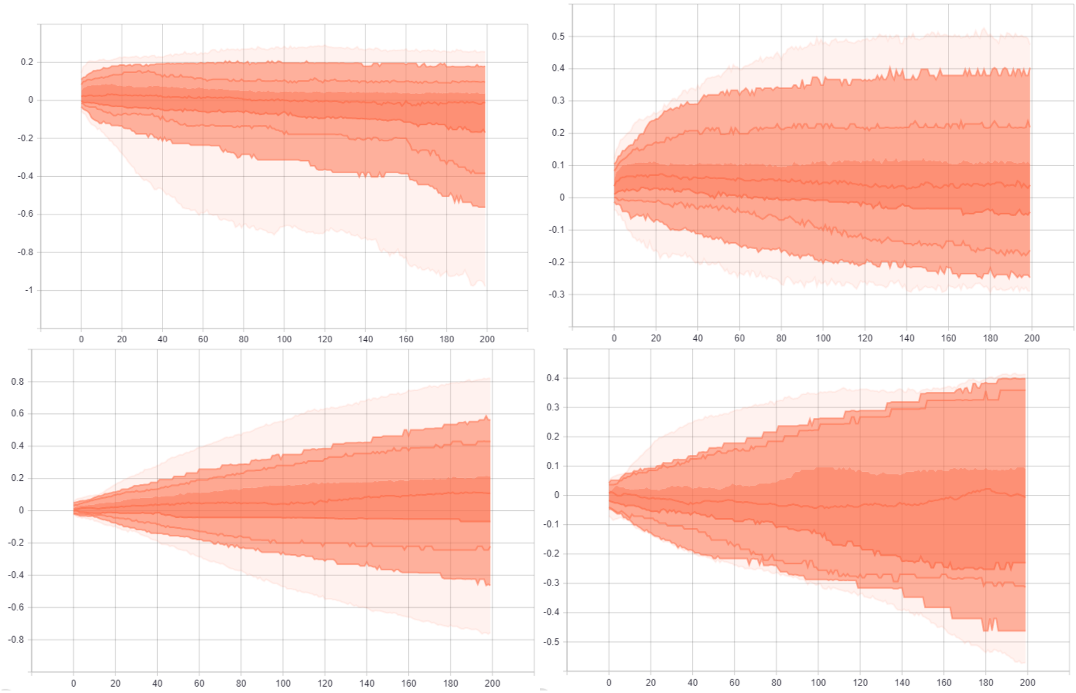
Waist data set Histogram of bias weights for the four layers of the network using ReLu at 200 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



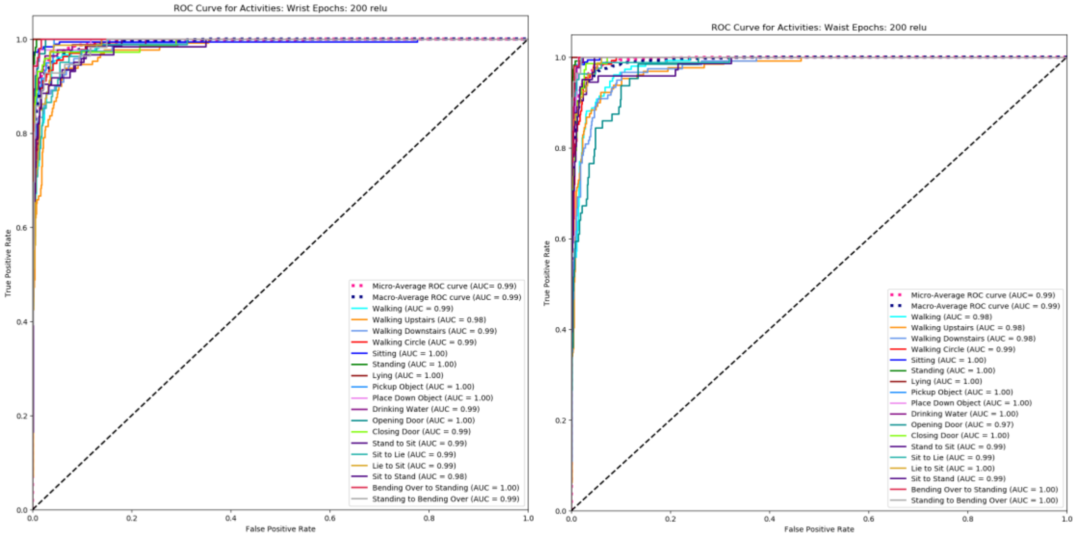
Waist data set Distribution of kernel weights for the four layers of the network using ReLu at 200 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



Waist data set Distribution of bias weights for the four layers of the network using ReLu at 200 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.

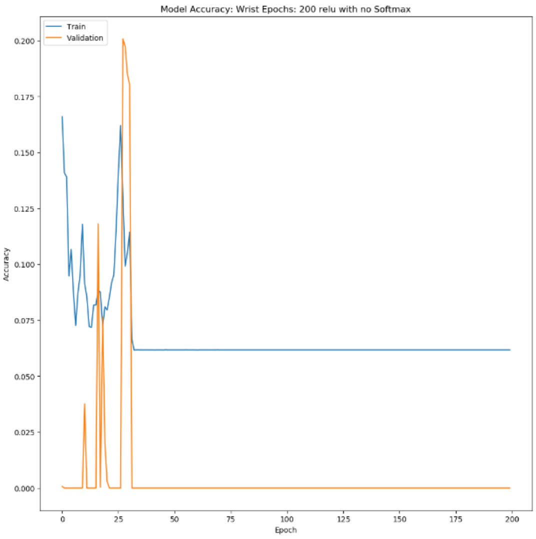


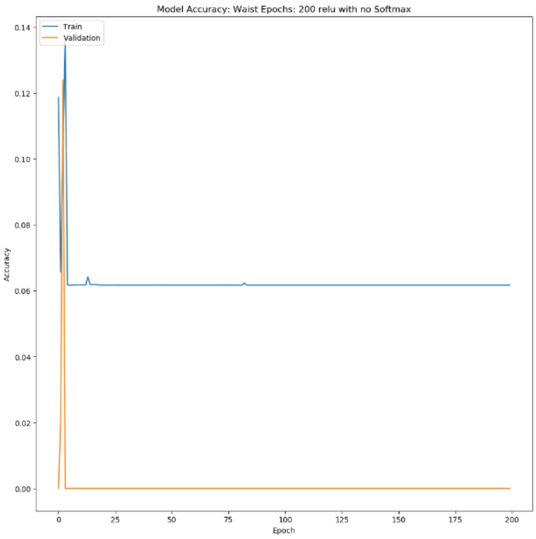
ROC Curve for wrist and waist data.



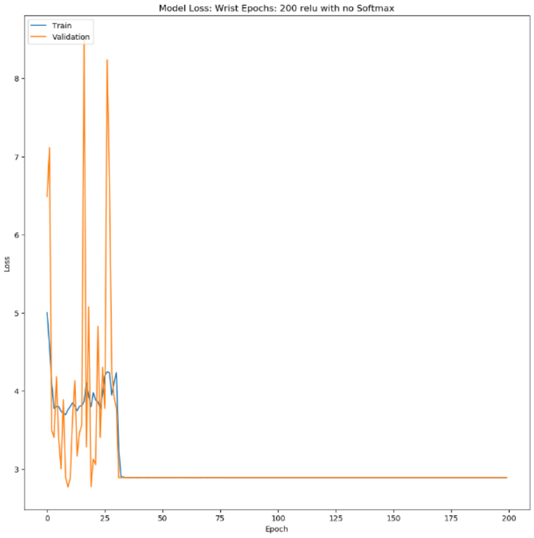
Appendix E – Additional Graphs: ReLu 200 Epochs No Softmax

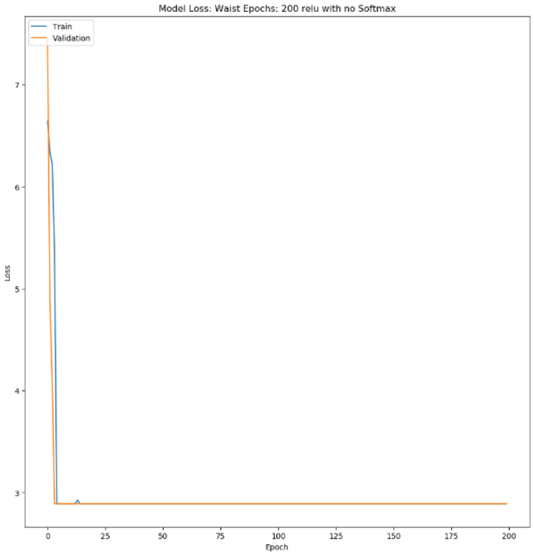
Wrist and Waist Accuracy



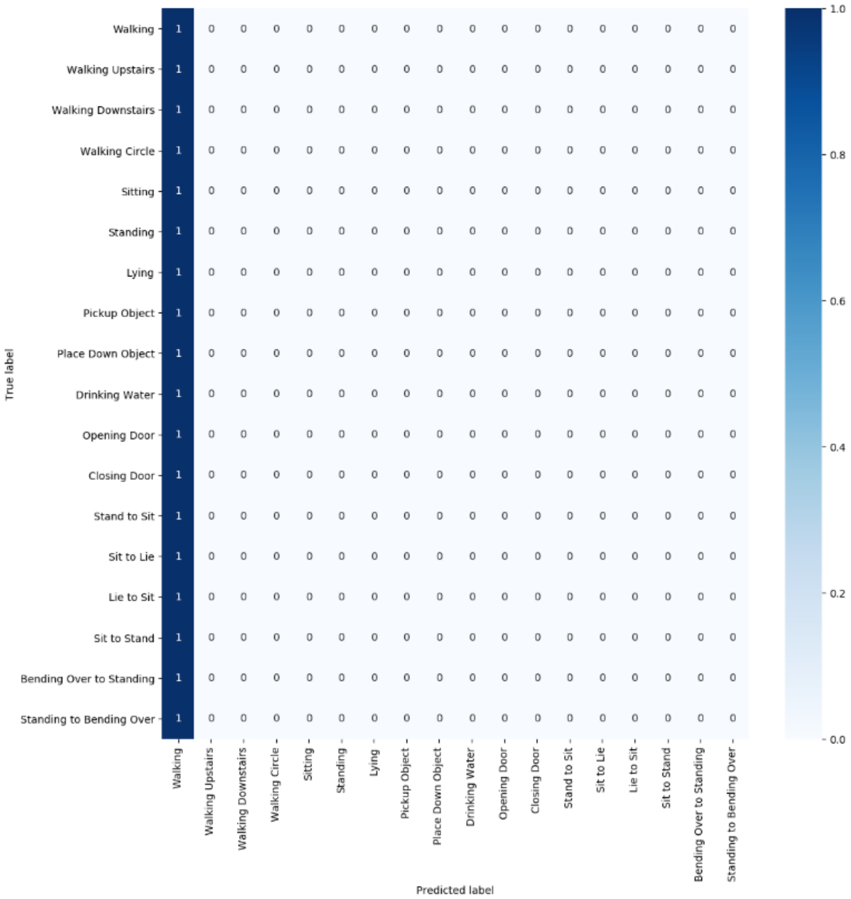


Wrist and Waist Loss

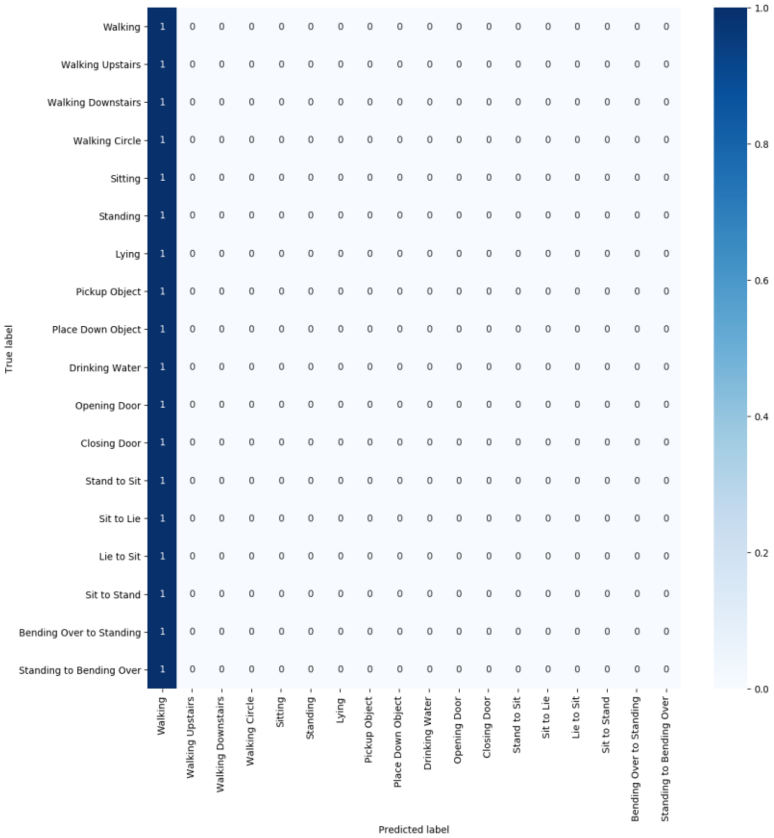




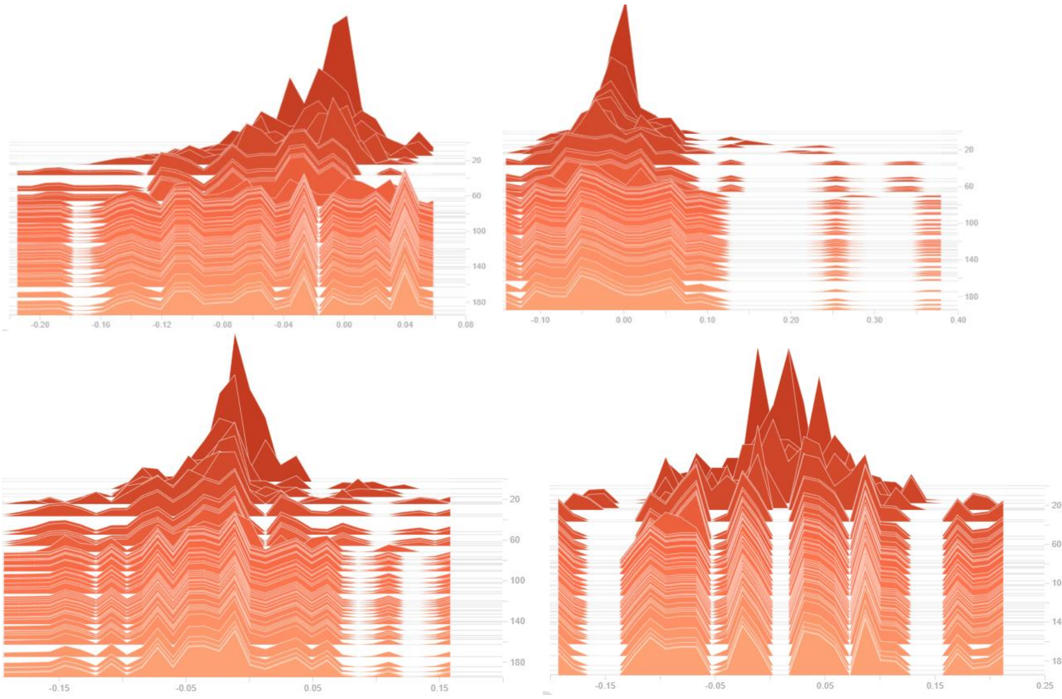
Wrist Confusion Matrix



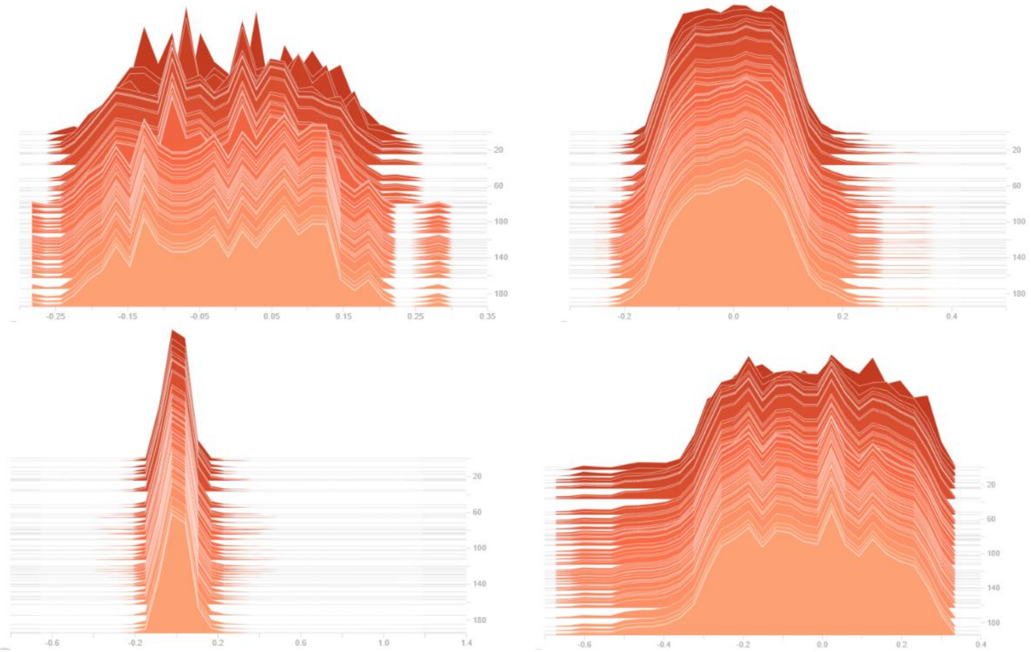
Waist Confusion Matrix



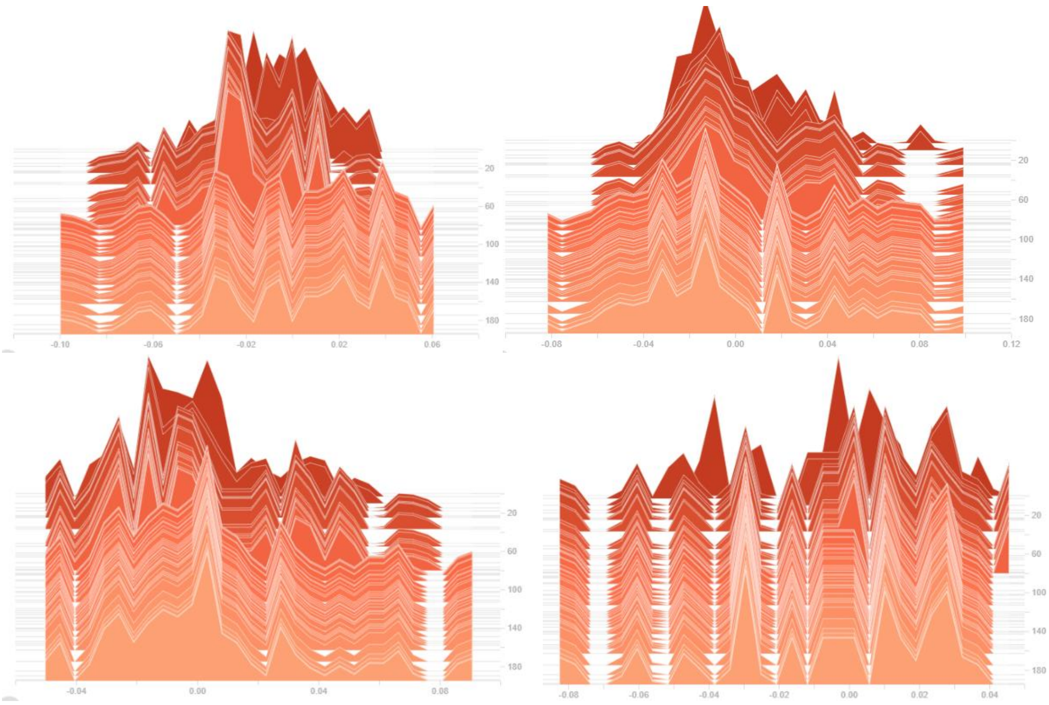
Wrist data set Histogram of bias weights for the four layers of the network using ReLu at 200 epochs with no Softmax. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



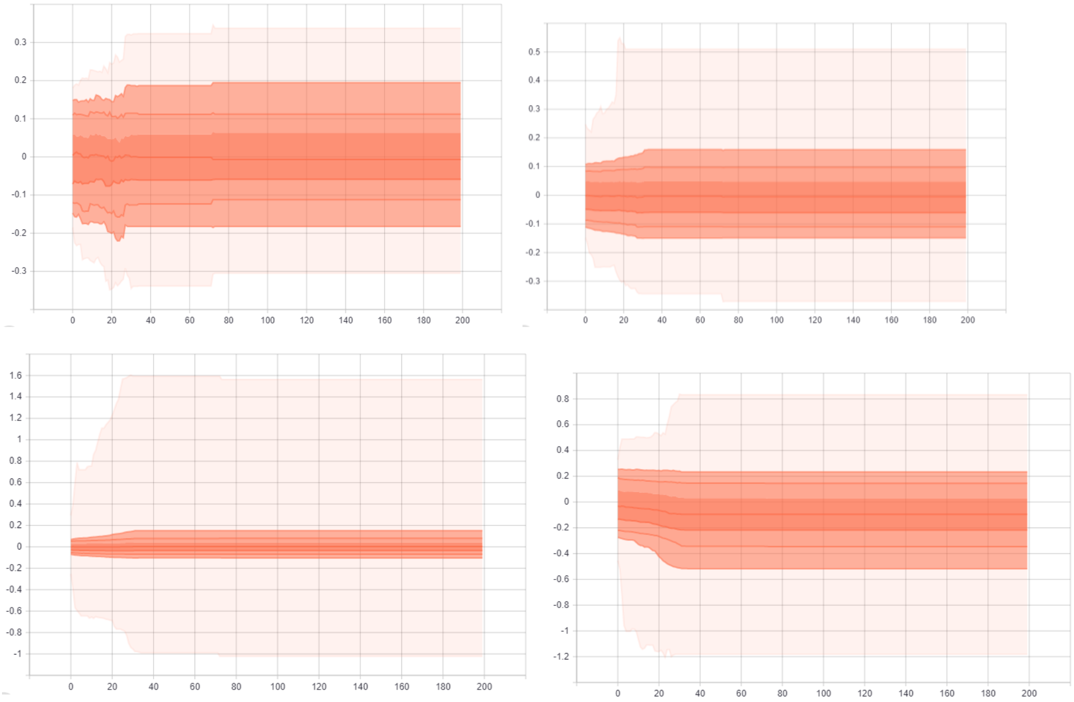
Waist data set Histogram of kernel weights for the four layers of the network using ReLu at 200 epochs with no Softmax. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



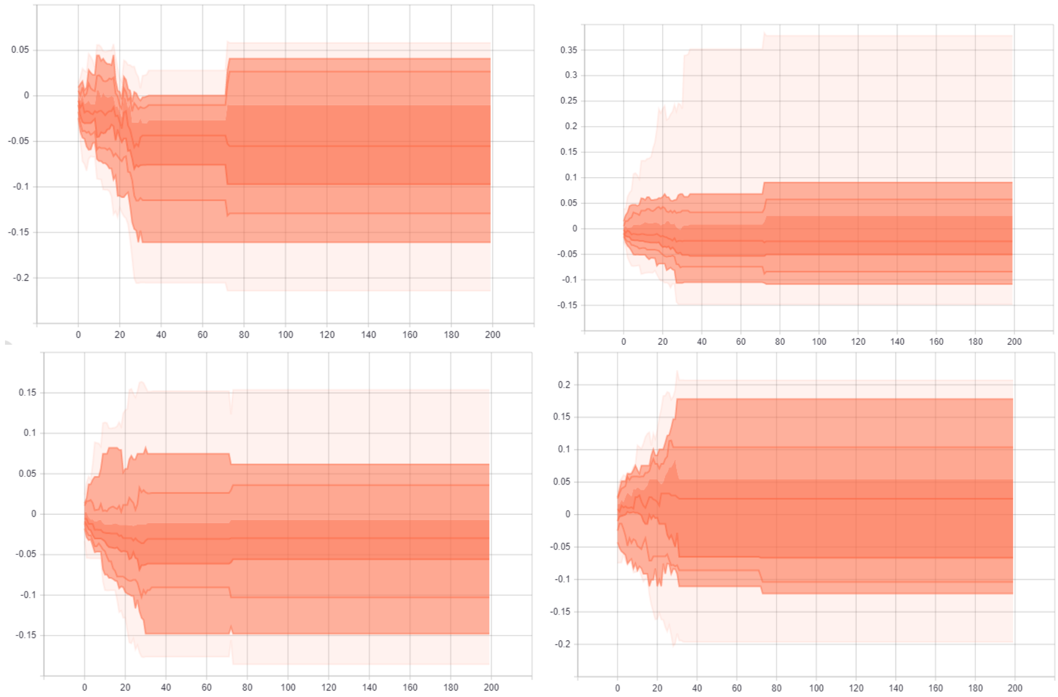
Waist data set Histogram of bias weights for the four layers of the network using ReLu at 200 epochs with no Softmax. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



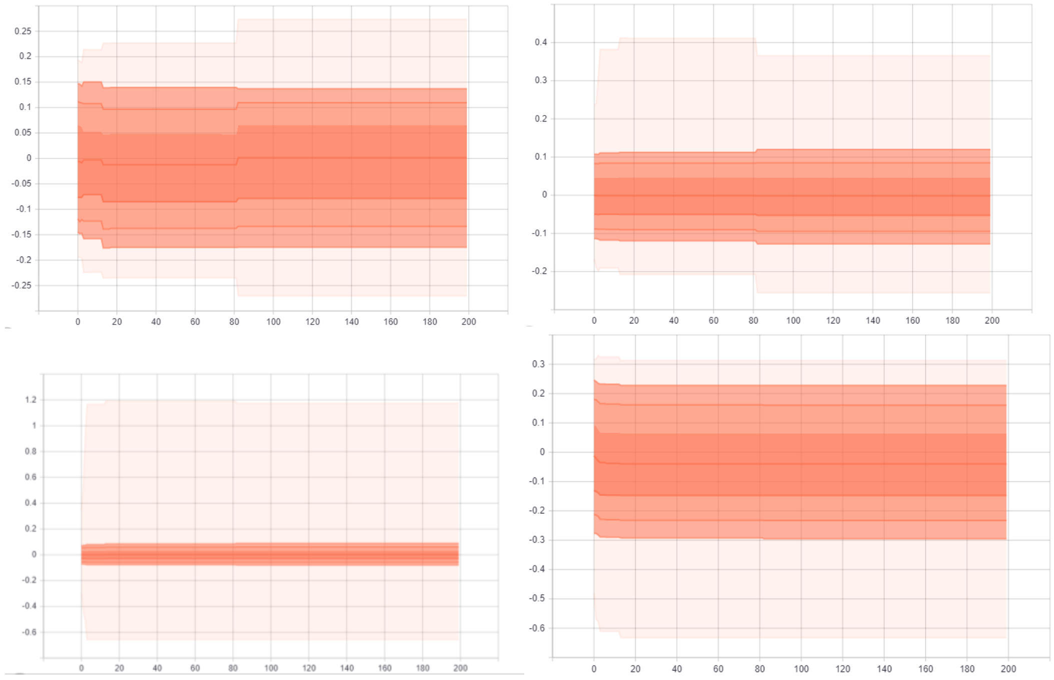
Wrist data set Distribution of kernel weights for the four layers of the network using ReLu at 200 epochs with no Softmax. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



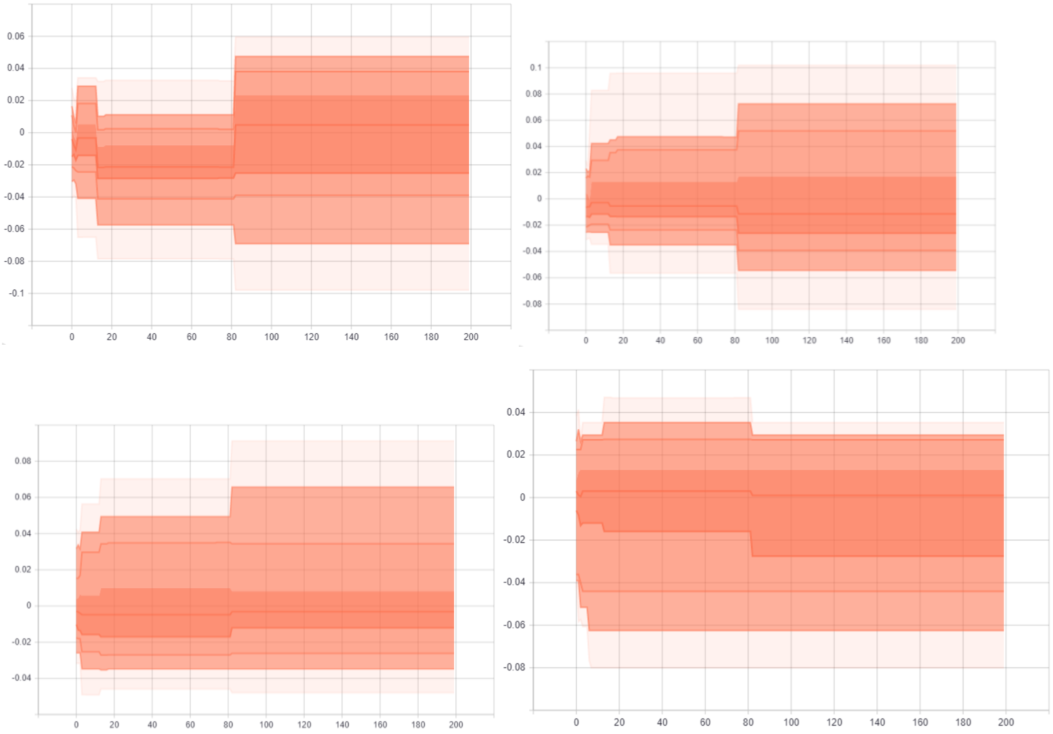
Wrist data set Distribution of bias weights for the four layers of the network using ReLu at 200 epochs with no Softmax. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



Waist data set Distribution of kernel weights for the four layers of the network using ReLu at 200 epochs with no Softmax. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.

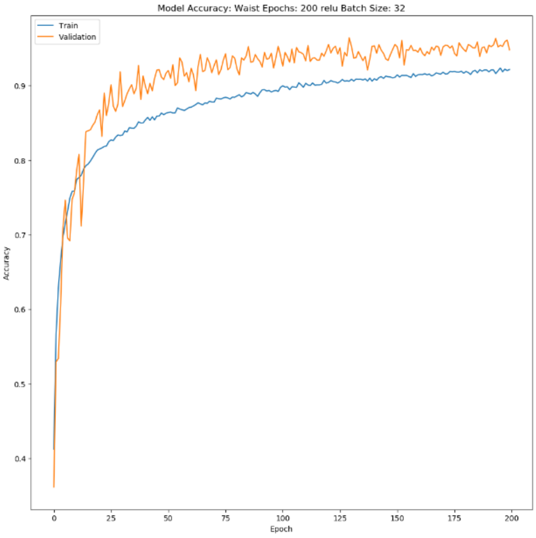


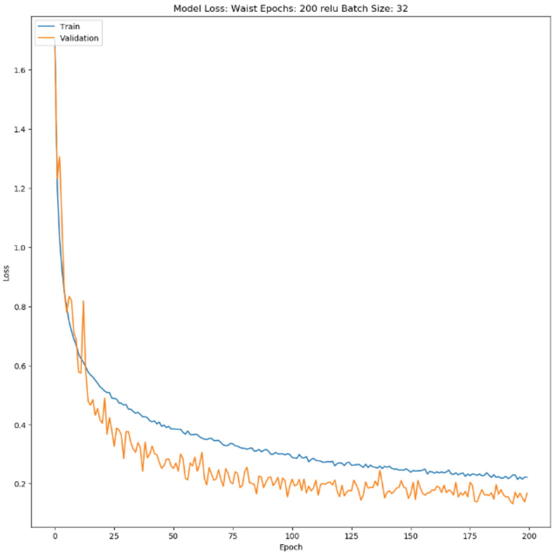
Waist data set Distribution of bias weights for the four layers of the network using ReLu at 200 epochs with no Softmax. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



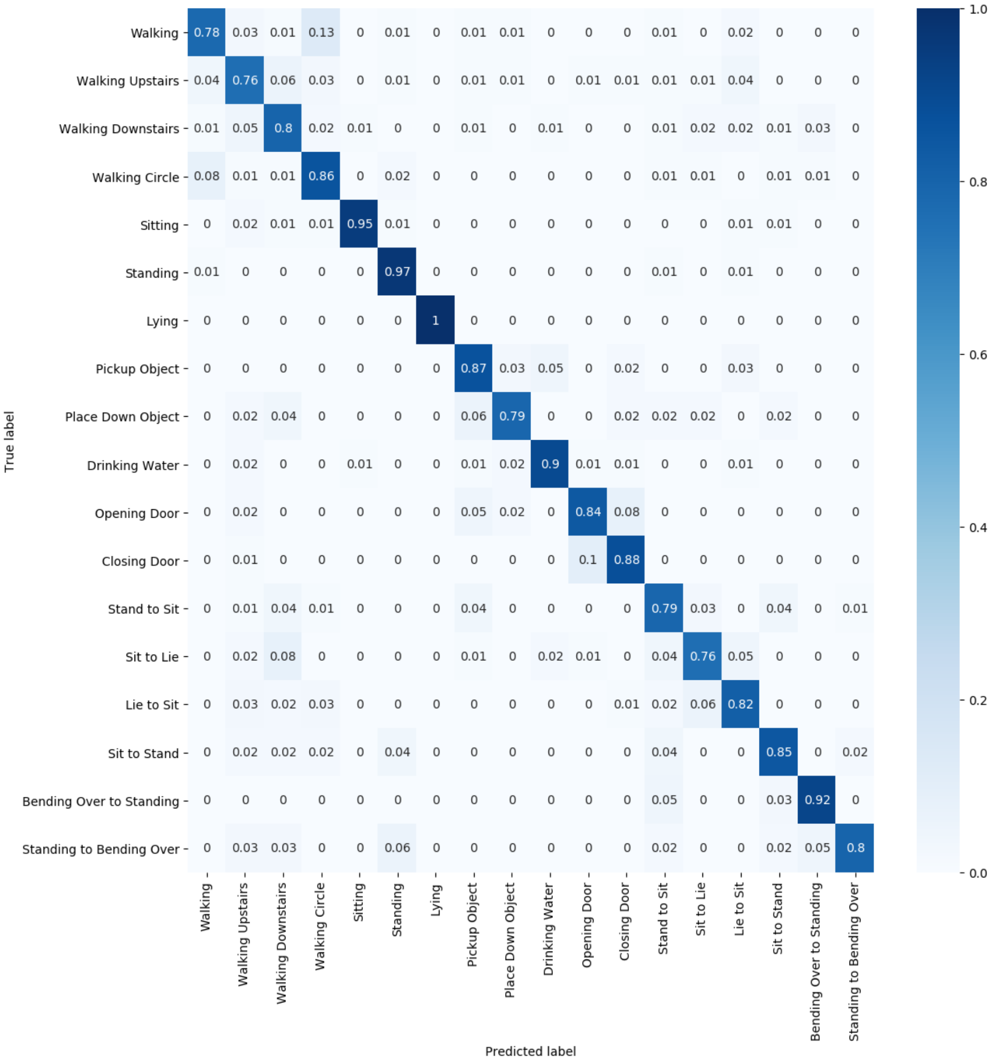
Appendix F – Additional Graphs: ReLu 200 Epochs Batch Size 32

Waist Accuracy and Loss

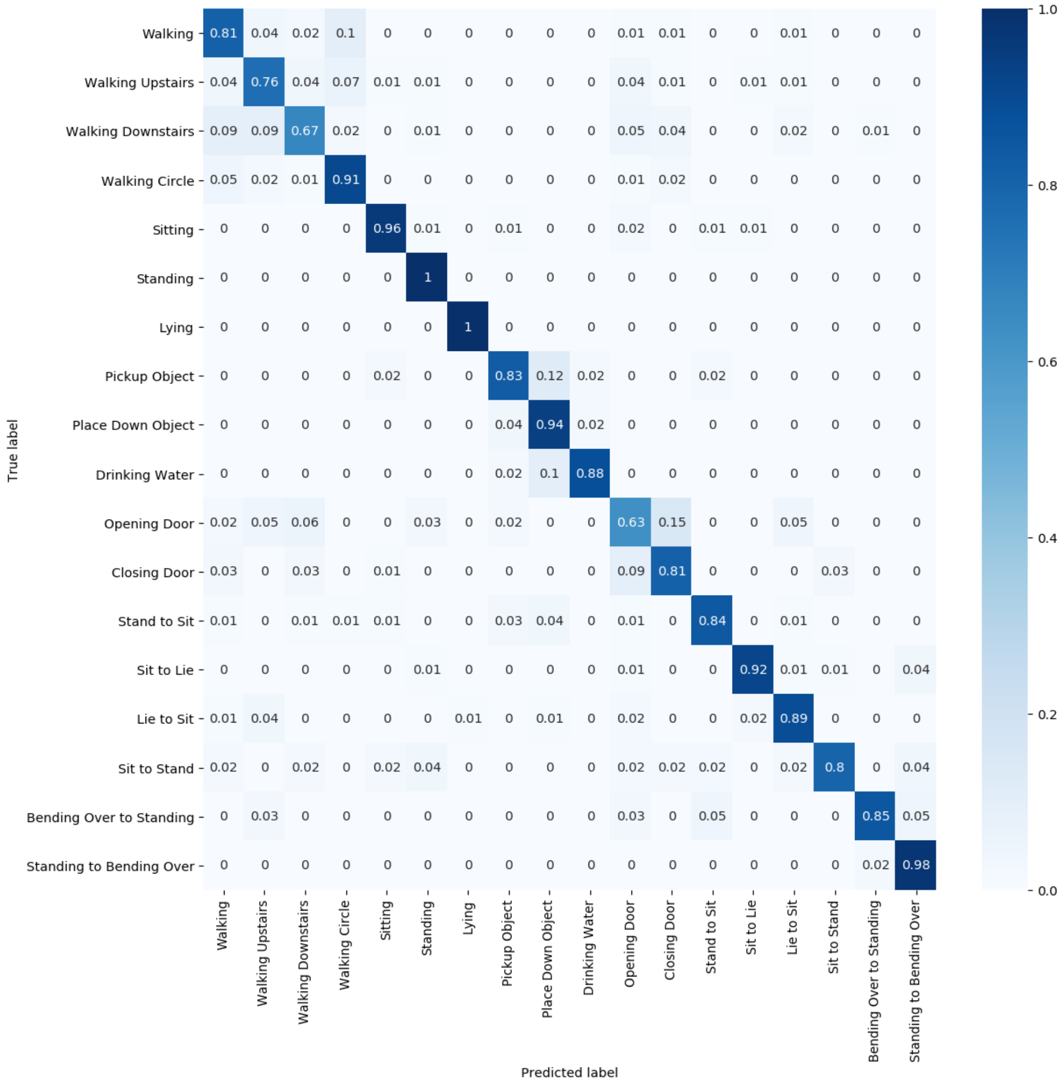




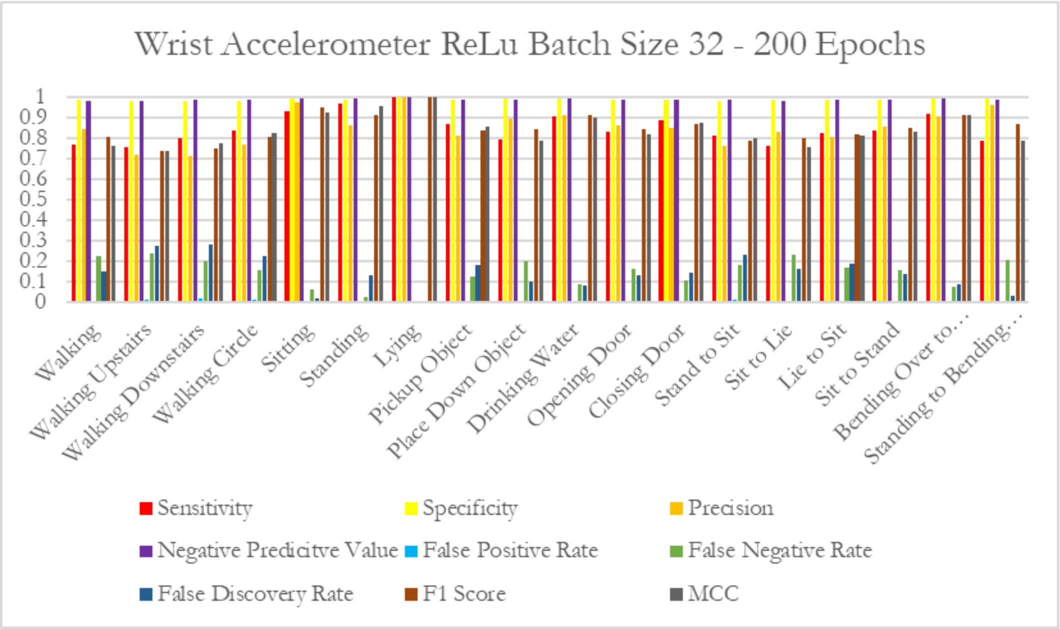
Confusion matrix for wrist and waist

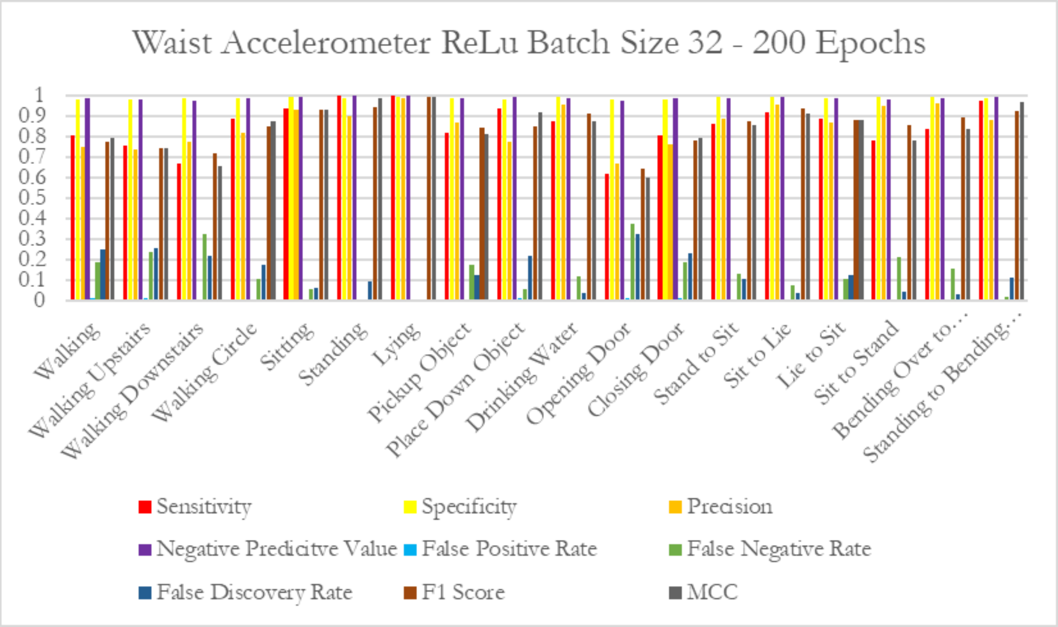


	Walking	Walking Upstairs	Walking Downstairs	Walking Circle	Sitting	Standing	Lying	Pickup Object	Place Down Object	Drinking Water	Opening Door	Closing Door	Stand to Sit	Sit to Lie	Lie to Sit	Sit to Stand	Bending Over to Standing	Standing to Bending Over
Walking	0.78	0.03	0.01	0.13	0	0.01	0	0.01	0.01	0	0	0	0.01	0	0.02	0	0	0
Walking Upstairs	0.04	0.76	0.06	0.03	0	0.01	0	0.01	0.01	0	0.01	0.01	0.01	0.01	0.04	0	0	0
Walking Downstairs	0.01	0.05	0.8	0.02	0.01	0	0	0.01	0	0.01	0	0	0.01	0.02	0.02	0.01	0.03	0
Walking Circle	0.08	0.01	0.01	0.86	0	0.02	0	0	0	0	0	0	0.01	0.01	0	0.01	0.01	0
Sitting	0	0.02	0.01	0.01	0.95	0.01	0	0	0	0	0	0	0	0	0.01	0.01	0	0
Standing	0.01	0	0	0	0	0.97	0	0	0	0	0	0	0.01	0	0.01	0	0	0
Lying	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Pickup Object	0	0	0	0	0	0	0	0.87	0.03	0.05	0	0.02	0	0	0.03	0	0	0
Place Down Object	0	0.02	0.04	0	0	0	0	0.06	0.79	0	0	0.02	0.02	0.02	0	0.02	0	0
Drinking Water	0	0.02	0	0	0.01	0	0	0.01	0.02	0.9	0.01	0.01	0	0	0.01	0	0	0
Opening Door	0	0.02	0	0	0	0	0	0.05	0.02	0	0.84	0.08	0	0	0	0	0	0
Closing Door	0	0.01	0	0	0	0	0	0	0	0	0.1	0.88	0	0	0	0	0	0
Stand to Sit	0	0.01	0.04	0.01	0	0	0	0.04	0	0	0	0	0.79	0.03	0	0.04	0	0.01
Sit to Lie	0	0.02	0.08	0	0	0	0	0.01	0	0.02	0.01	0	0.04	0.76	0.05	0	0	0
Lie to Sit	0	0.03	0.02	0.03	0	0	0	0	0	0	0	0.01	0.02	0.06	0.82	0	0	0
Sit to Stand	0	0.02	0.02	0.02	0	0.04	0	0	0	0	0	0	0.04	0	0	0.85	0	0.02
Bending Over to Standing	0	0	0	0	0	0	0	0	0	0	0	0	0.05	0	0	0.03	0.92	0
Standing to Bending Over	0	0.03	0.03	0	0	0.06	0	0	0	0	0	0	0.02	0	0	0.02	0.05	0.8

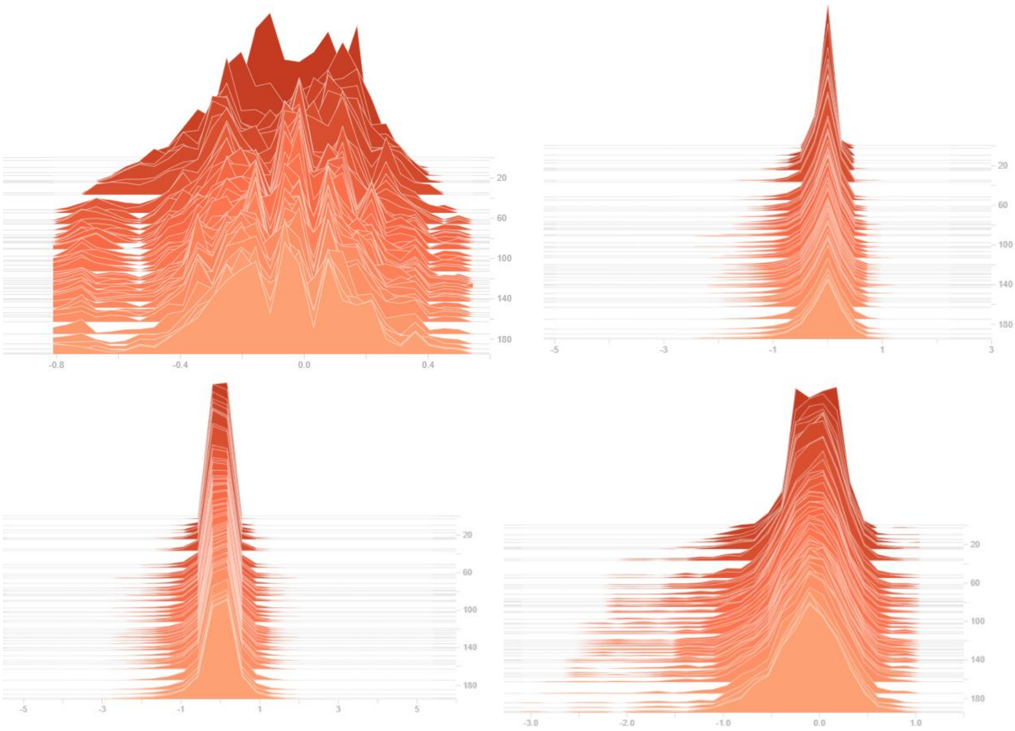


True label \ Predicted label	Walking	Walking Upstairs	Walking Downstairs	Walking Circle	Sitting	Standing	Lying	Pickup Object	Place Down Object	Drinking Water	Opening Door	Closing Door	Stand to Sit	Sit to Lie	Lie to Sit	Sit to Stand	Bending Over to Standing	Standing to Bending Over
Walking	0.81	0.04	0.02	0.1	0	0	0	0	0	0	0.01	0.01	0	0	0.01	0	0	0
Walking Upstairs	0.04	0.76	0.04	0.07	0.01	0.01	0	0	0	0	0.04	0.01	0	0.01	0.01	0	0	0
Walking Downstairs	0.09	0.09	0.67	0.02	0	0.01	0	0	0	0	0.05	0.04	0	0	0.02	0	0.01	0
Walking Circle	0.05	0.02	0.01	0.91	0	0	0	0	0	0	0.01	0.02	0	0	0	0	0	0
Sitting	0	0	0	0	0.96	0.01	0	0.01	0	0	0.02	0	0.01	0.01	0	0	0	0
Standing	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Lying	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Pickup Object	0	0	0	0	0.02	0	0	0.83	0.12	0.02	0	0	0.02	0	0	0	0	0
Place Down Object	0	0	0	0	0	0	0	0.04	0.94	0.02	0	0	0	0	0	0	0	0
Drinking Water	0	0	0	0	0	0	0	0.02	0.1	0.88	0	0	0	0	0	0	0	0
Opening Door	0.02	0.05	0.06	0	0	0.03	0	0.02	0	0	0.63	0.15	0	0	0.05	0	0	0
Closing Door	0.03	0	0.03	0	0.01	0	0	0	0	0	0.09	0.81	0	0	0	0.03	0	0
Stand to Sit	0.01	0	0.01	0.01	0.01	0	0	0.03	0.04	0	0.01	0	0.84	0	0.01	0	0	0
Sit to Lie	0	0	0	0	0	0.01	0	0	0	0	0.01	0	0	0.92	0.01	0.01	0	0.04
Lie to Sit	0.01	0.04	0	0	0	0	0.01	0	0.01	0	0.02	0	0	0.02	0.89	0	0	0
Sit to Stand	0.02	0	0.02	0	0.02	0.04	0	0	0	0	0.02	0.02	0.02	0	0.02	0.8	0	0.04
Bending Over to Standing	0	0.03	0	0	0	0	0	0	0	0	0.03	0	0.05	0	0	0	0.85	0.05
Standing to Bending Over	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.02	0.98

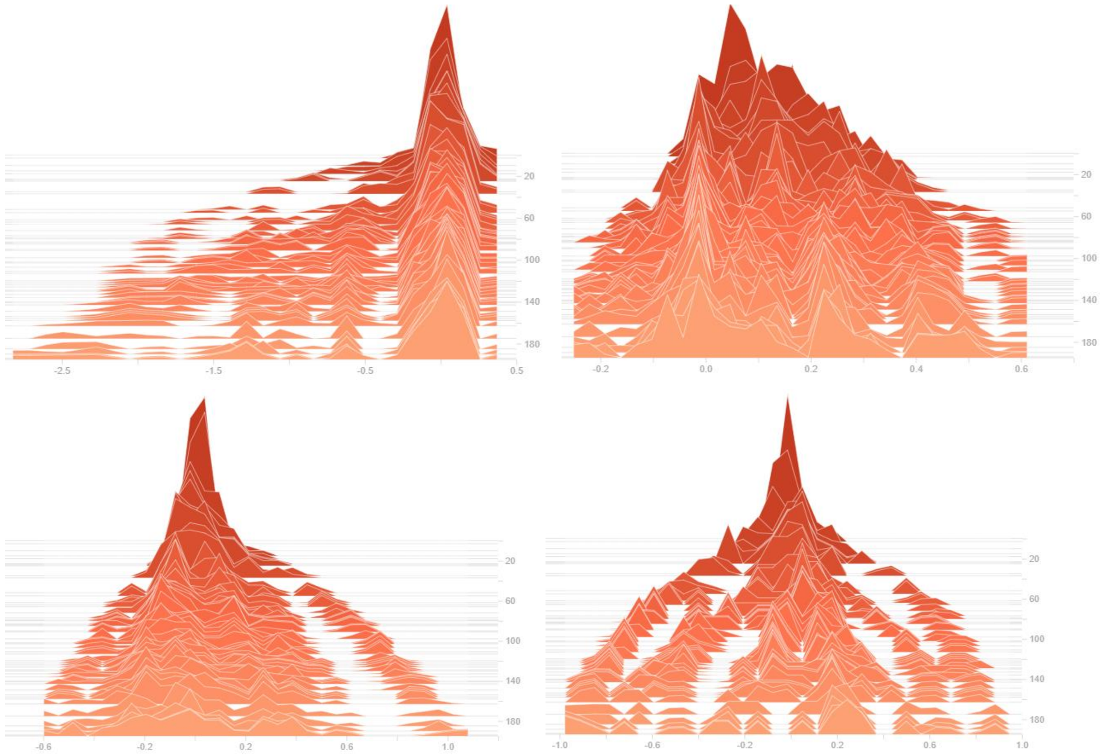




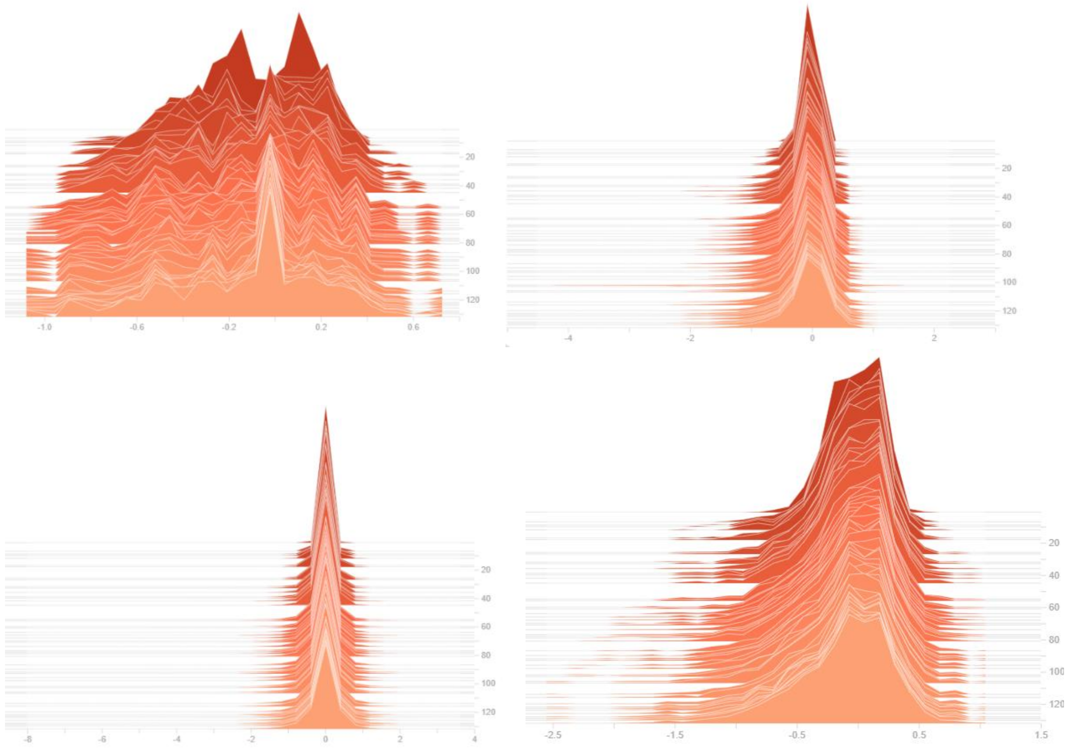
Wrist data set Histogram of kernel weights for the four layers of the network using ReLu at 200 epochs with batch size 32. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



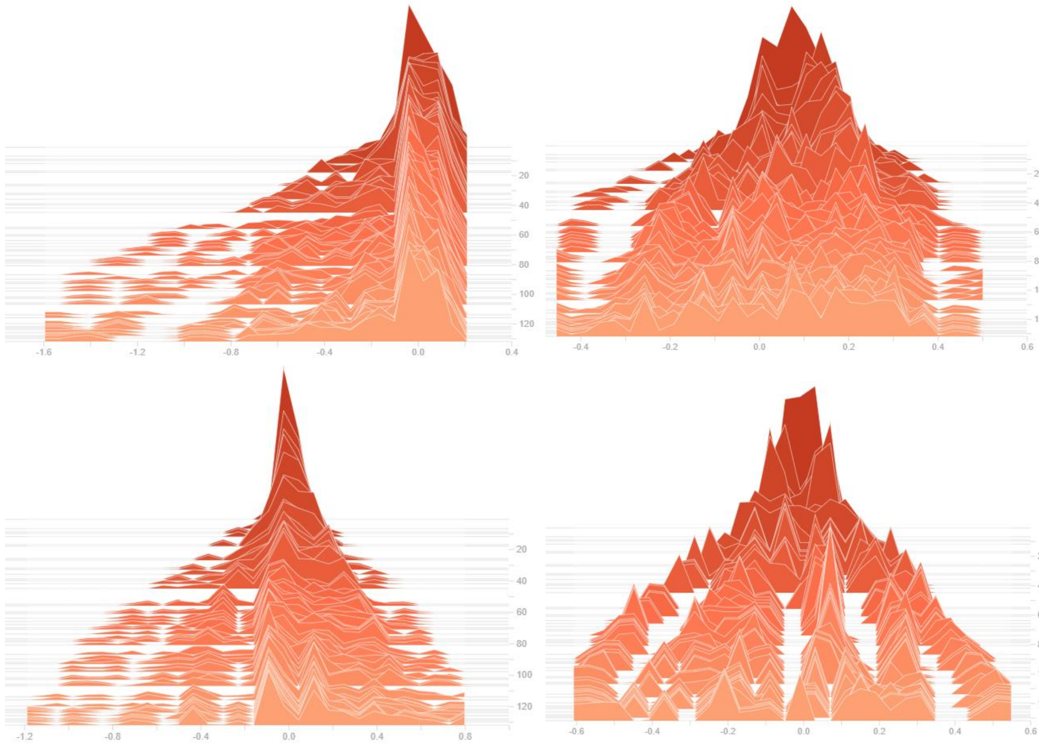
Wrist data set Histogram of bias weights for the four layers of the network using ReLu at 200 epochs with batch size 32. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



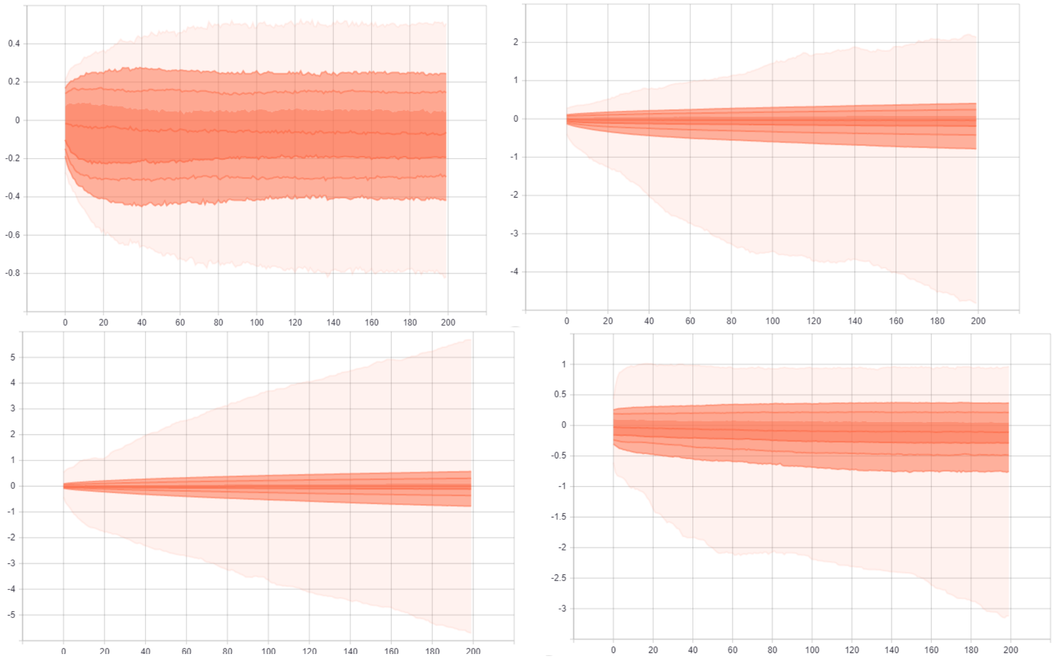
Waist data set Histogram of kernel weights for the four layers of the network using ReLu at 200 epochs with batch size 32. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



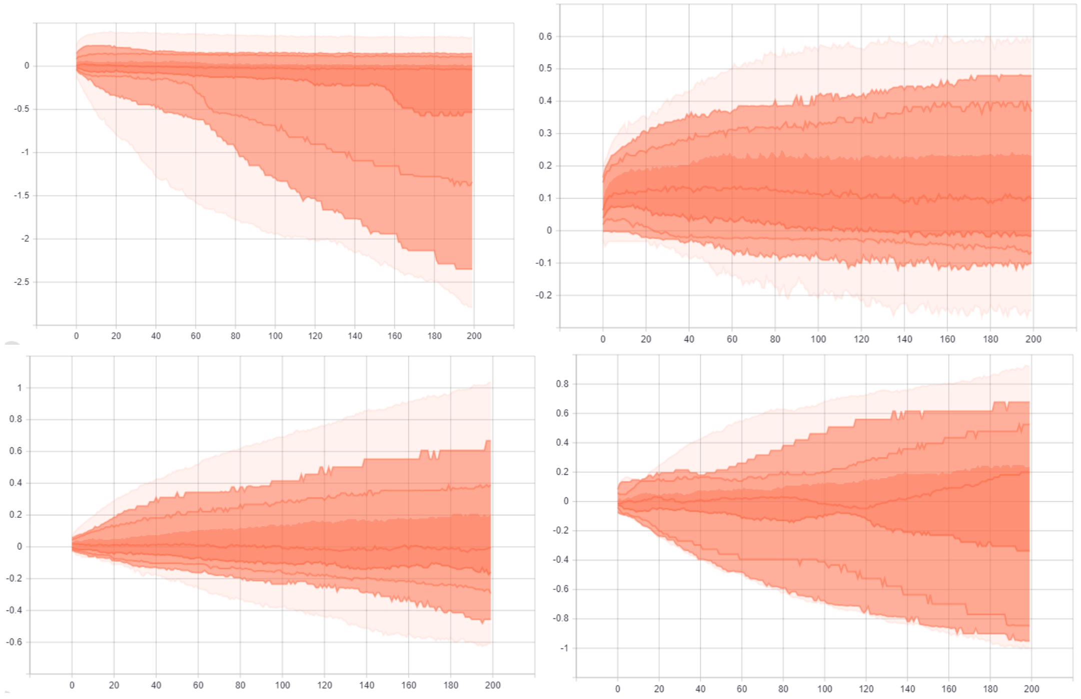
Waist data set Histogram of bias weights for the four layers of the network using ReLu at 200 epochs with batch size 32. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



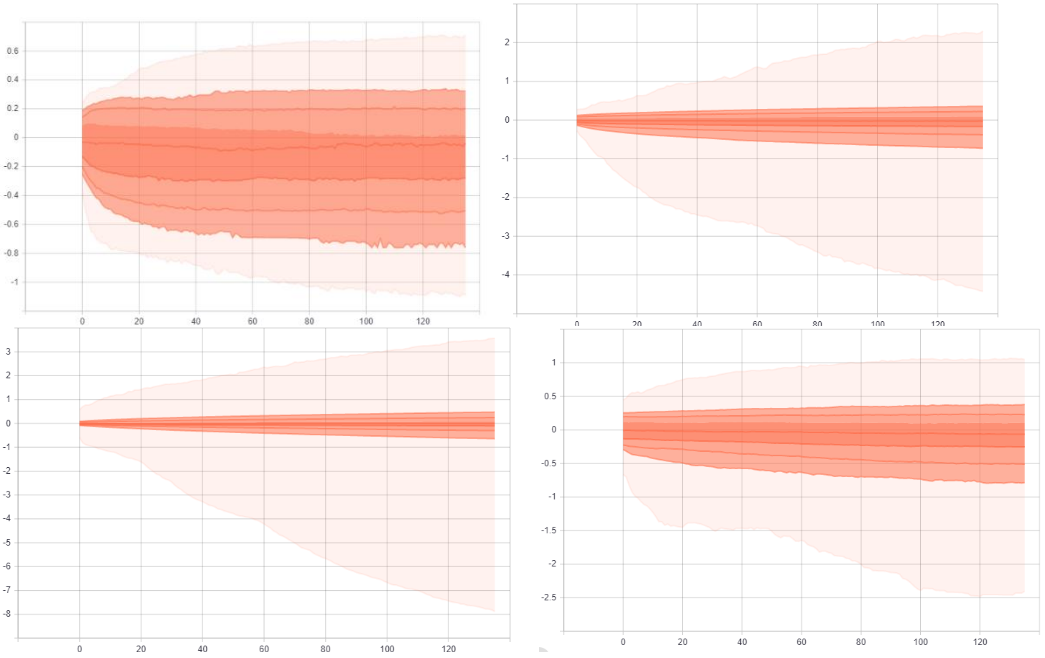
Wrist data set Distribution of kernel weights for the four layers of the network using ReLu at 200 epochs with batch size 32. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



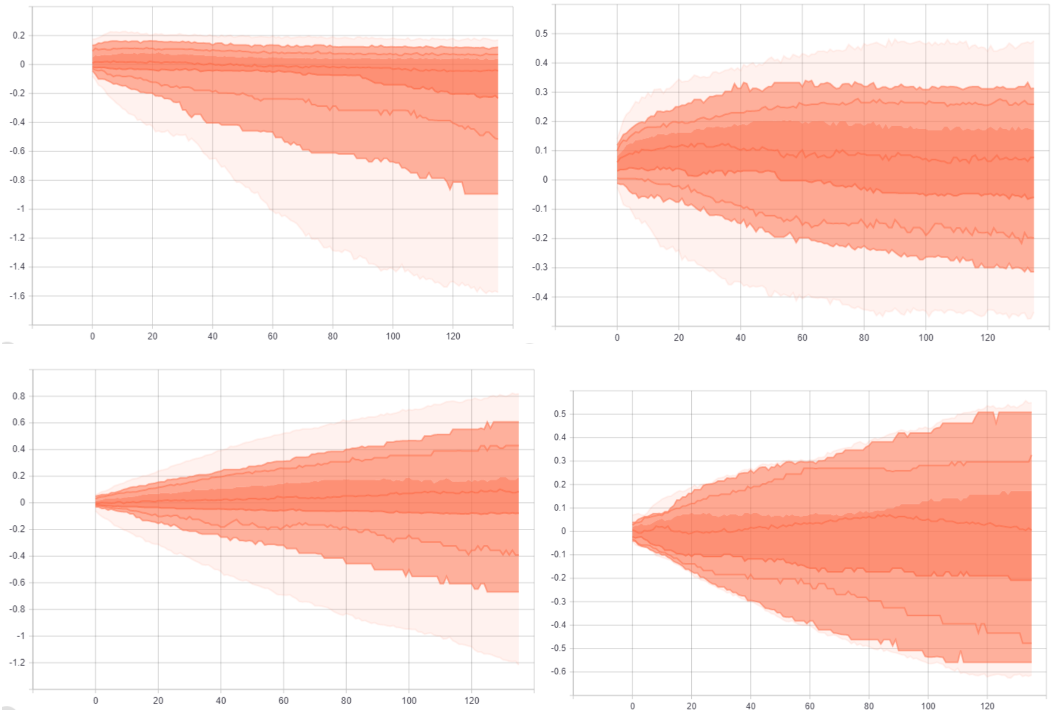
Wrist data set Distribution of bias weights for the four layers of the network using ReLu at 200 epochs with batch size 32. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



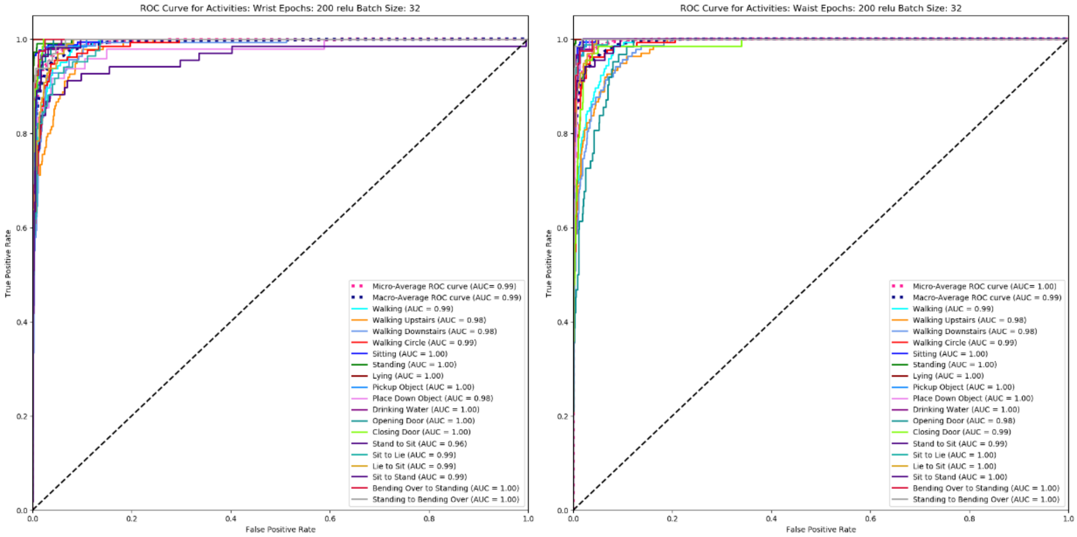
Waist data set Distribution of kernel weights for the four layers of the network using ReLu at 200 epochs with batch size 32. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



Waist data set Distribution of bias weights for the four layers of the network using ReLu at 200 epochs with batch size 32. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.

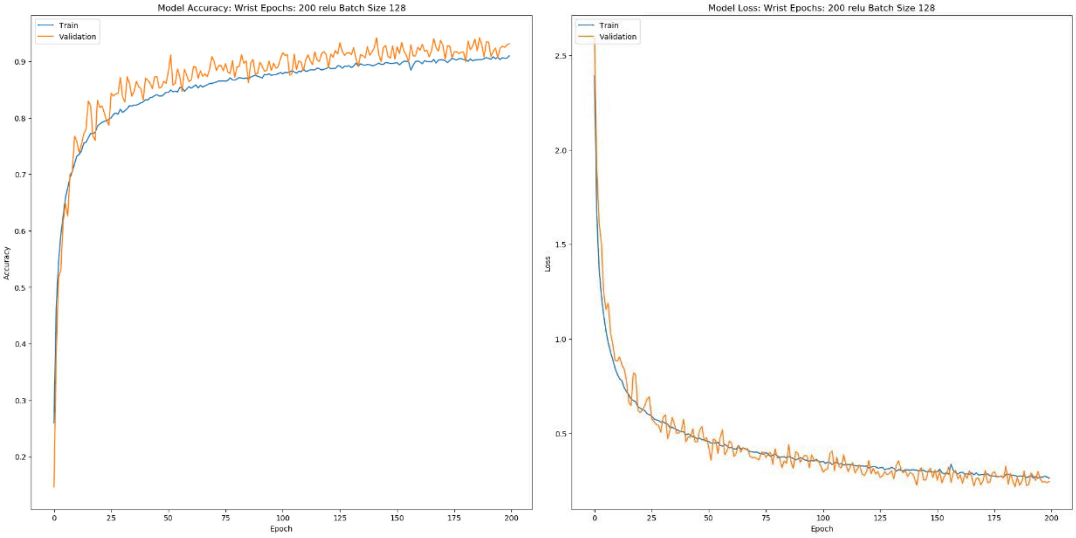


ROC Curve for Wrist and Waist

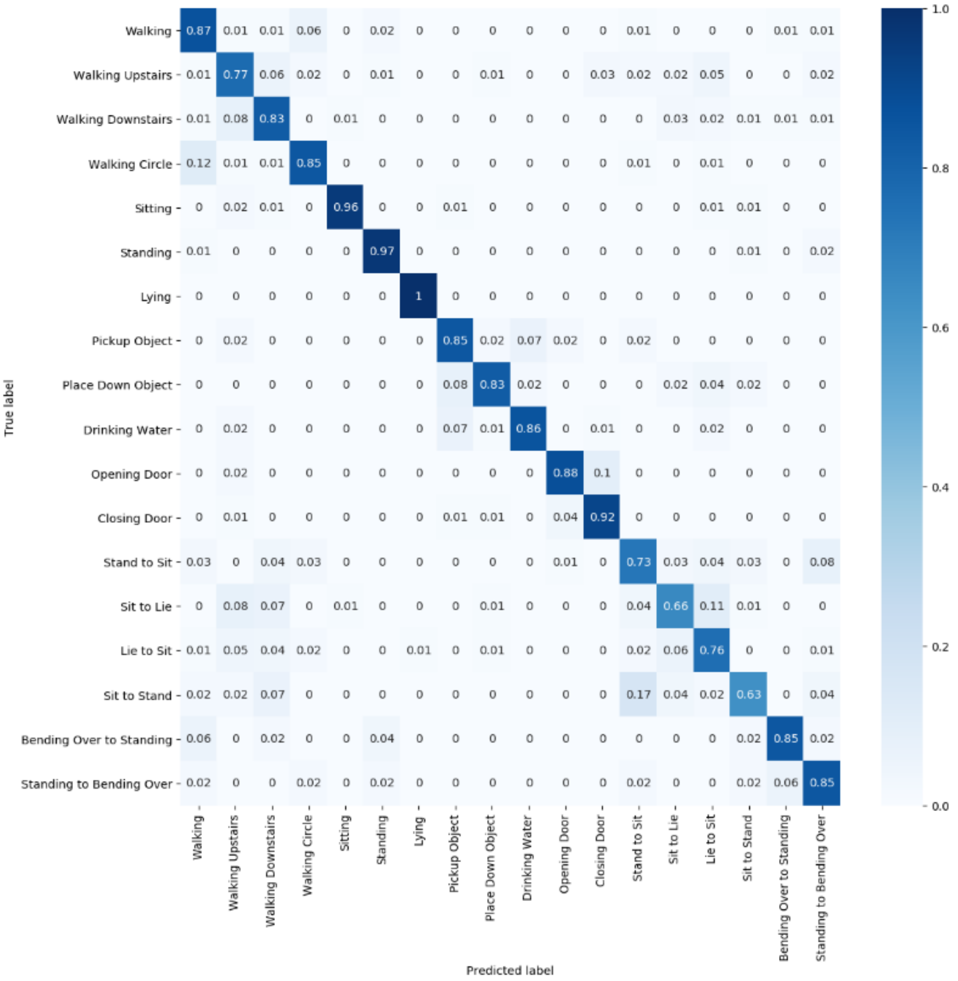


Appendix G – Additional Graphs: ReLu 200 Epochs Batch Size 128

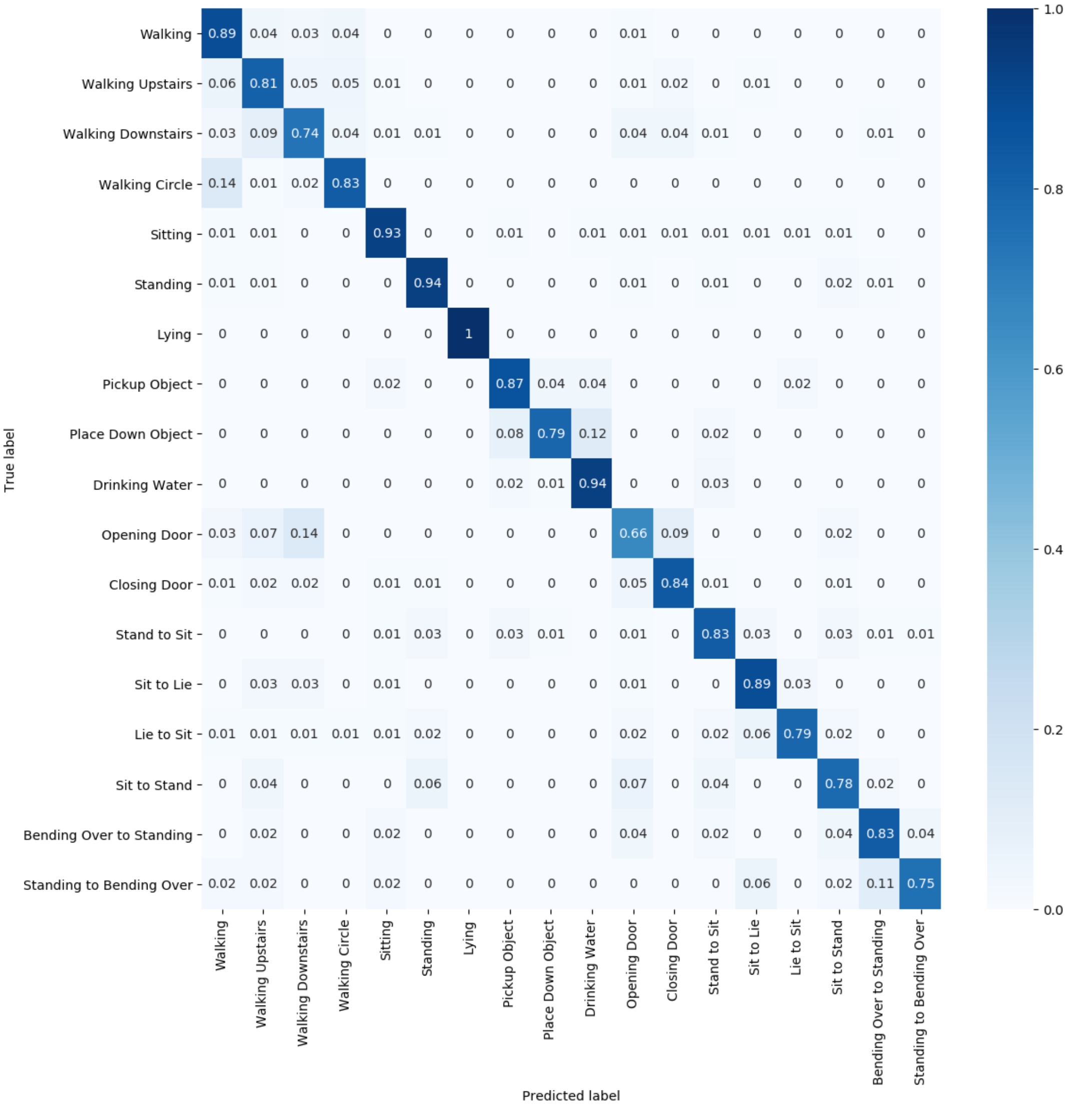
Wrist accuracy and loss

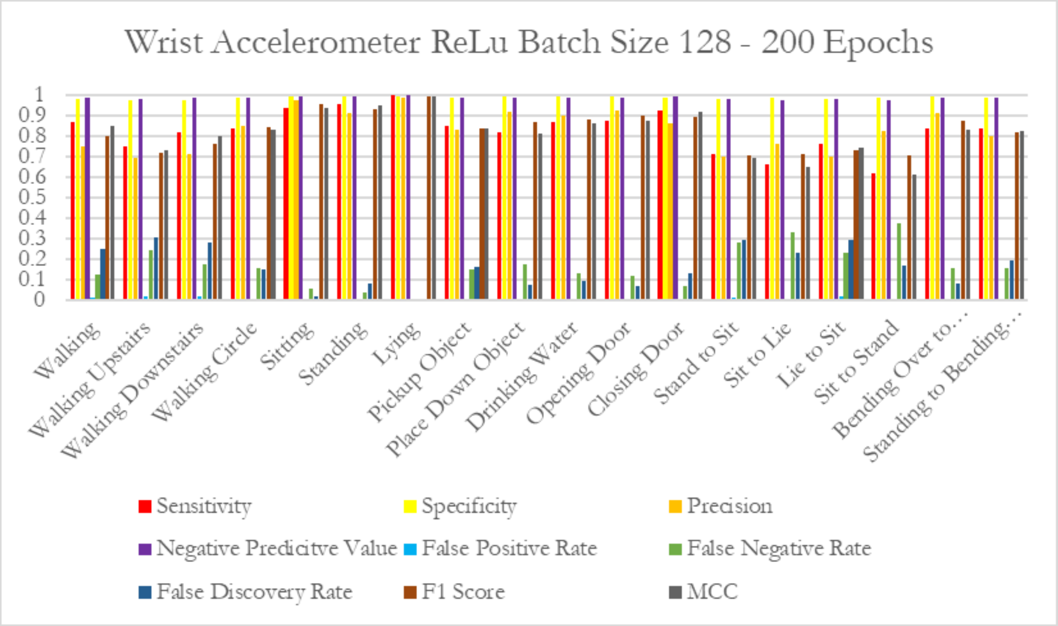


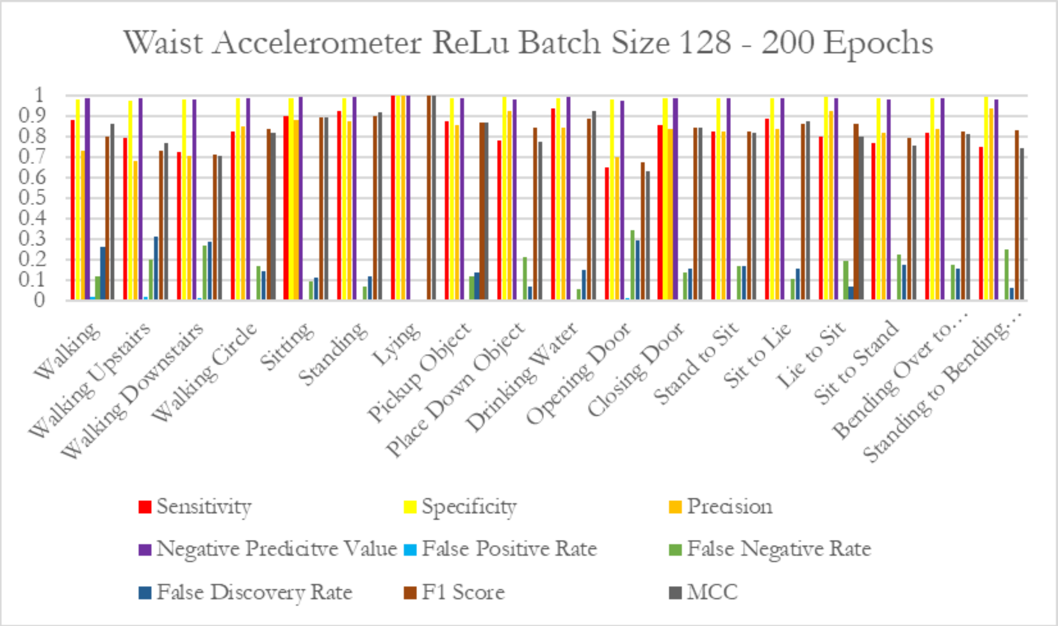
Wrist and waist confusion matrix



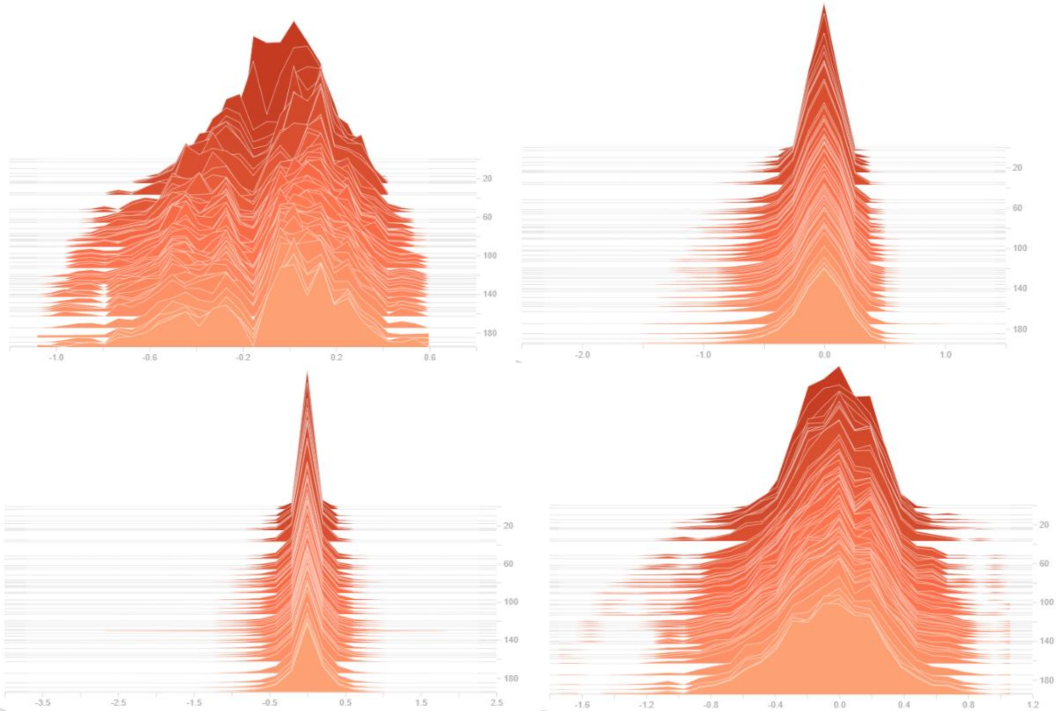
	Walking	Walking Upstairs	Walking Downstairs	Walking Circle	Sitting	Standing	Lying	Pickup Object	Place Down Object	Drinking Water	Opening Door	Closing Door	Stand to Sit	Sit to Lie	Lie to Sit	Sit to Stand	Bending Over to Standing	Standing to Bending Over
Walking	0.87	0.01	0.01	0.06	0	0.02	0	0	0	0	0	0	0.01	0	0	0	0.01	0.01
Walking Upstairs	0.01	0.77	0.06	0.02	0	0.01	0	0	0.01	0	0	0.03	0.02	0.02	0.05	0	0	0.02
Walking Downstairs	0.01	0.08	0.83	0	0.01	0	0	0	0	0	0	0	0	0.03	0.02	0.01	0.01	0.01
Walking Circle	0.12	0.01	0.01	0.85	0	0	0	0	0	0	0	0	0.01	0	0.01	0	0	0
Sitting	0	0.02	0.01	0	0.96	0	0	0.01	0	0	0	0	0	0	0.01	0.01	0	0
Standing	0.01	0	0	0	0	0.97	0	0	0	0	0	0	0	0	0	0.01	0	0.02
Lying	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Pickup Object	0	0.02	0	0	0	0	0	0.85	0.02	0.07	0.02	0	0.02	0	0	0	0	0
Place Down Object	0	0	0	0	0	0	0	0.08	0.83	0.02	0	0	0	0.02	0.04	0.02	0	0
Drinking Water	0	0.02	0	0	0	0	0	0.07	0.01	0.86	0	0.01	0	0	0.02	0	0	0
Opening Door	0	0.02	0	0	0	0	0	0	0	0	0.88	0.1	0	0	0	0	0	0
Closing Door	0	0.01	0	0	0	0	0	0.01	0.01	0	0.04	0.92	0	0	0	0	0	0
Stand to Sit	0.03	0	0.04	0.03	0	0	0	0	0	0	0.01	0	0.73	0.03	0.04	0.03	0	0.08
Sit to Lie	0	0.08	0.07	0	0.01	0	0	0	0.01	0	0	0	0.04	0.66	0.11	0.01	0	0
Lie to Sit	0.01	0.05	0.04	0.02	0	0	0.01	0	0.01	0	0	0	0.02	0.06	0.76	0	0	0.01
Sit to Stand	0.02	0.02	0.07	0	0	0	0	0	0	0	0	0	0.17	0.04	0.02	0.63	0	0.04
Bending Over to Standing	0.06	0	0.02	0	0	0.04	0	0	0	0	0	0	0	0	0	0.02	0.85	0.02
Standing to Bending Over	0.02	0	0	0.02	0	0.02	0	0	0	0	0	0	0.02	0	0	0.02	0.06	0.85


True label
Walking 0.89 0.04 0.03 0.04 0 0 0 0 0 0 0.01 0 0 0 0 0 0 0
Walking Upstairs 0.06 0.81 0.05 0.05 0.01 0 0 0 0 0 0.01 0.02 0 0.01 0 0 0 0
Walking Downstairs 0.03 0.09 0.74 0.04 0.01 0.01 0 0 0 0 0.04 0.04 0.01 0 0 0 0.01 0
Walking Circle 0.14 0.01 0.02 0.83 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Sitting 0.01 0.01 0 0 0.93 0 0 0.01 0 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0 0
Standing 0.01 0.01 0 0 0 0.94 0 0 0 0 0.01 0 0.01 0 0 0.02 0.01 0
Lying 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
Pickup Object 0 0 0 0 0.02 0 0 0.87 0.04 0.04 0 0 0 0 0.02 0 0 0
Place Down Object 0 0 0 0 0 0 0 0.08 0.79 0.12 0 0 0.02 0 0 0 0 0
Drinking Water 0 0 0 0 0 0 0 0.02 0.01 0.94 0 0 0.03 0 0 0 0 0
Opening Door 0.03 0.07 0.14 0 0 0 0 0 0 0 0.66 0.09 0 0 0 0.02 0 0
Closing Door 0.01 0.02 0.02 0 0.01 0.01 0 0 0 0 0.05 0.84 0.01 0 0 0.01 0 0
Stand to Sit 0 0 0 0 0.01 0.03 0 0.03 0.01 0 0.01 0 0.83 0.03 0 0.03 0.01 0.01
Sit to Lie 0 0.03 0.03 0 0.01 0 0 0 0 0 0.01 0 0 0.89 0.03 0 0 0
Lie to Sit 0.01 0.01 0.01 0.01 0.01 0.02 0 0 0 0 0.02 0 0.02 0.06 0.79 0.02 0 0
Sit to Stand 0 0.04 0 0 0 0.06 0 0 0 0 0.07 0 0.04 0 0 0.78 0.02 0
Bending Over to Standing 0 0.02 0 0 0.02 0 0 0 0 0 0.04 0 0.02 0 0 0.04 0.83 0.04
Standing to Bending Over 0.02 0.02 0 0 0.02 0 0 0 0 0 0 0 0 0.06 0 0.02 0.11 0.75
Walking
Walking Upstairs
Walking Downstairs
Walking Circle
Sitting
Standing
Lying
Pickup Object
Place Down Object
Drinking Water
Opening Door
Closing Door
Stand to Sit
Sit to Lie
Lie to Sit
Sit to Stand
Bending Over to Standing
Standing to Bending Over
Predicted label
1.0
0.8
0.6
0.4
0.2
0.0

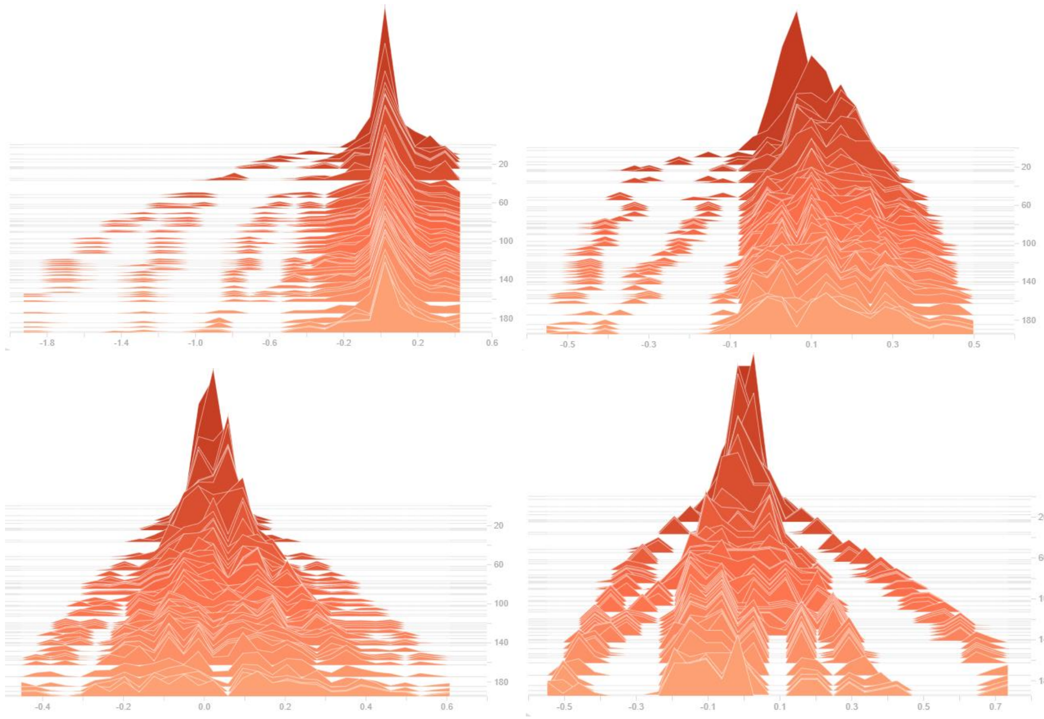

Wrist Accelerometer ReLu Batch Size 128 - 200 Epochs
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Walking
Walking Upstairs
Walking Downstairs
Walking Circle
Sitting
Standing
Lying
Pickup Object
Place Down Object
Drinking Water
Opening Door
Closing Door
Stand to Sit
Sit to Lie
Lie to Sit
Sit to Stand
Bending Over to...
Standing to Bending...
Sensitivity
Specificity
Precision
Negative Predicitve Value
False Positive Rate
False Negative Rate
False Discovery Rate
F1 Score
MCC



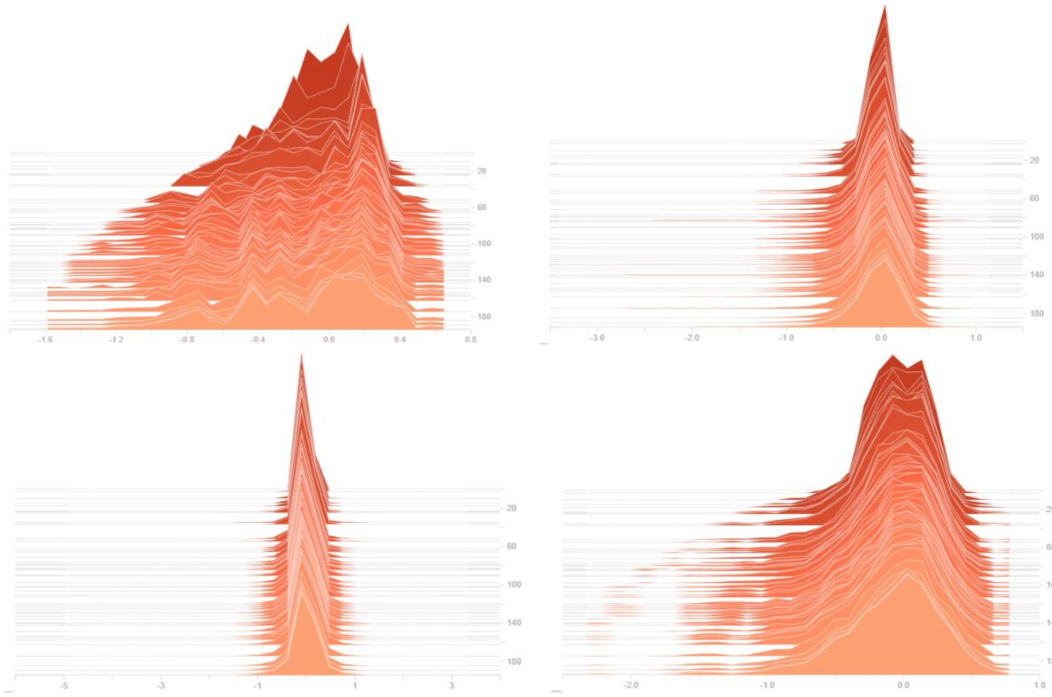
Wrist data set Histogram of kernel weights for the four layers of the network using ReLu at 200 epochs with batch size 128. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



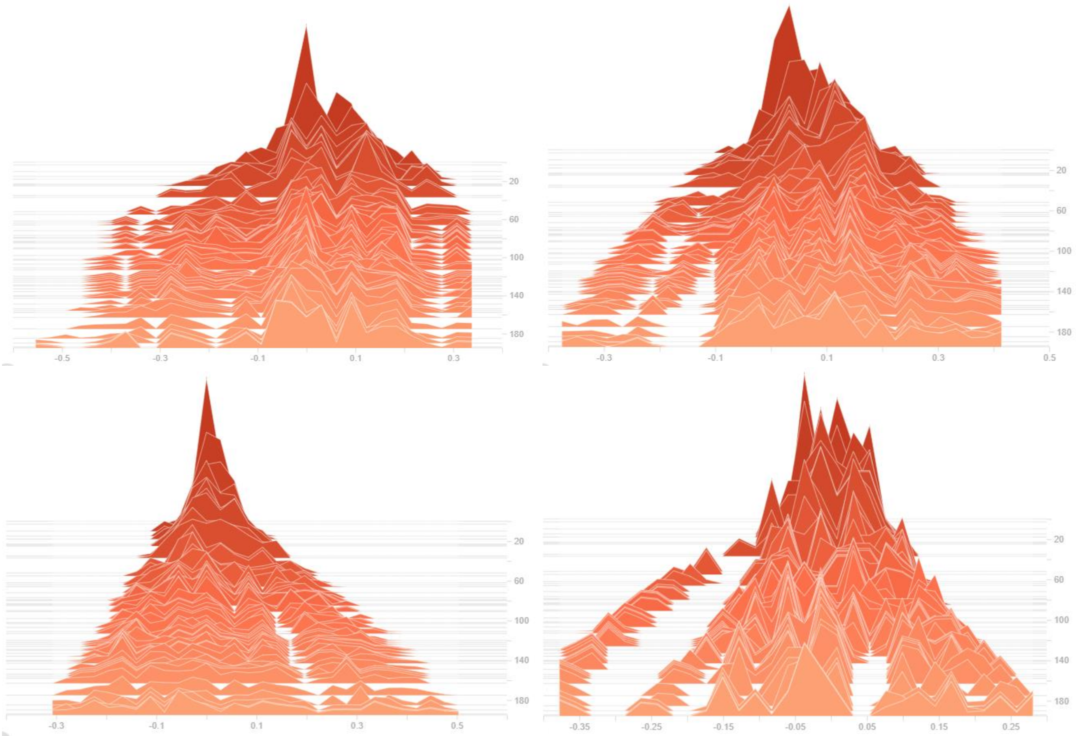
Wrist data set Histogram of bias weights for the four layers of the network using ReLu at 200 epochs with batch size 128. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



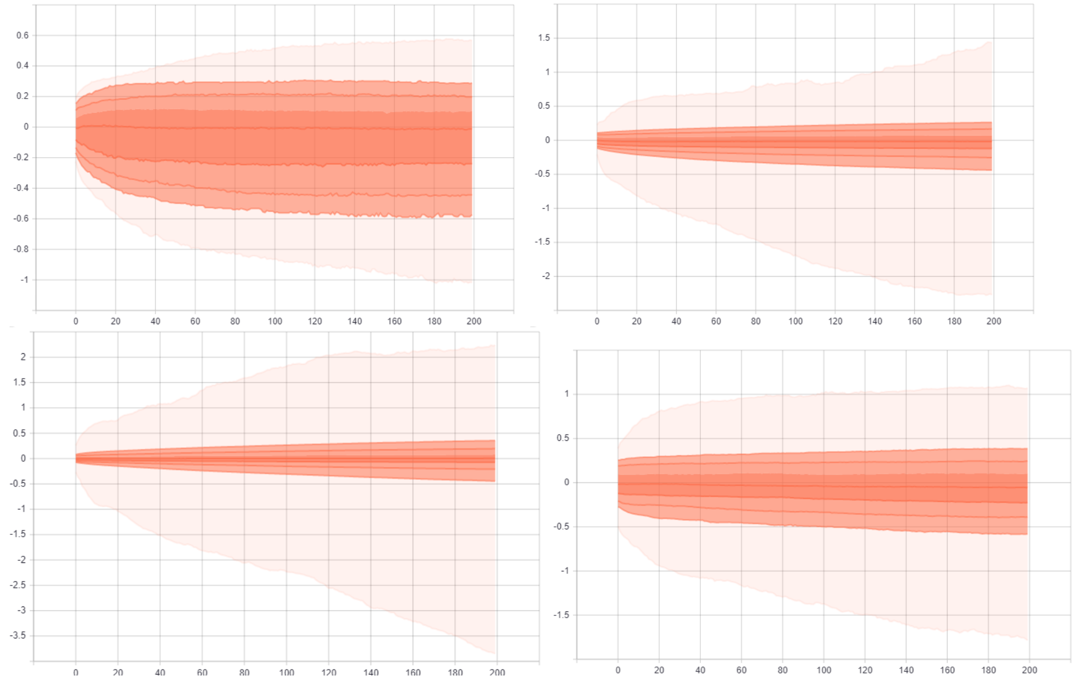
Waist data set Histogram of kernel weights for the four layers of the network using ReLu at 200 epochs with batch size 128. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



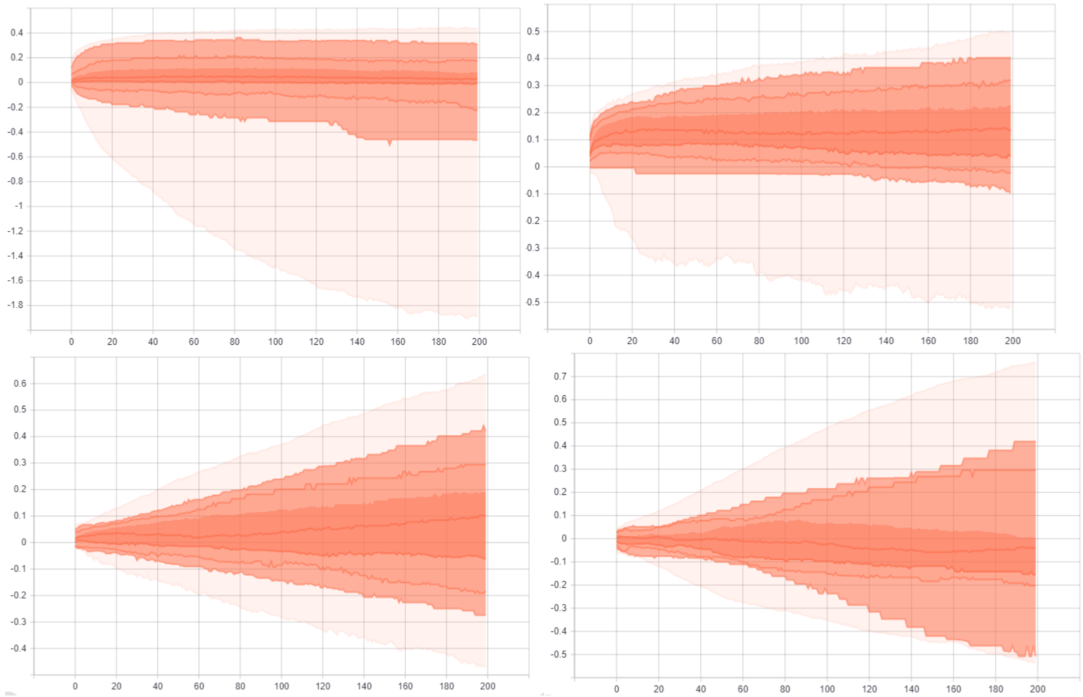
Waist data set Histogram of bias weights for the four layers of the network using ReLu at 200 epochs with batch size 128. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



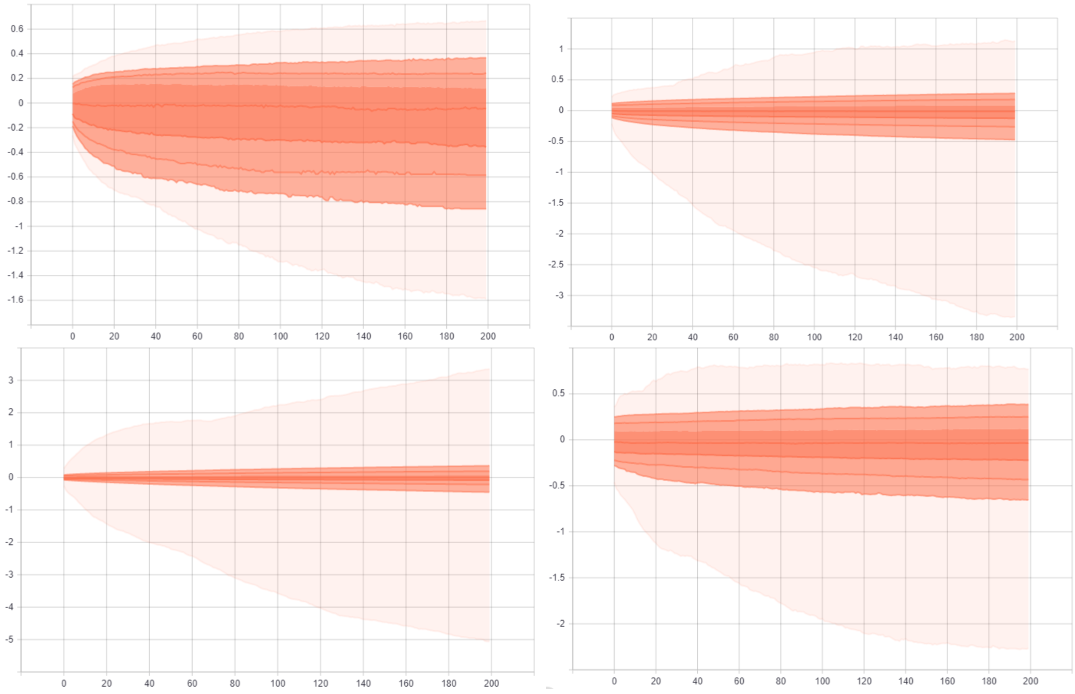
Wrist data set Distribution of kernel weights for the four layers of the network using ReLu at 200 epochs with batch size 128. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



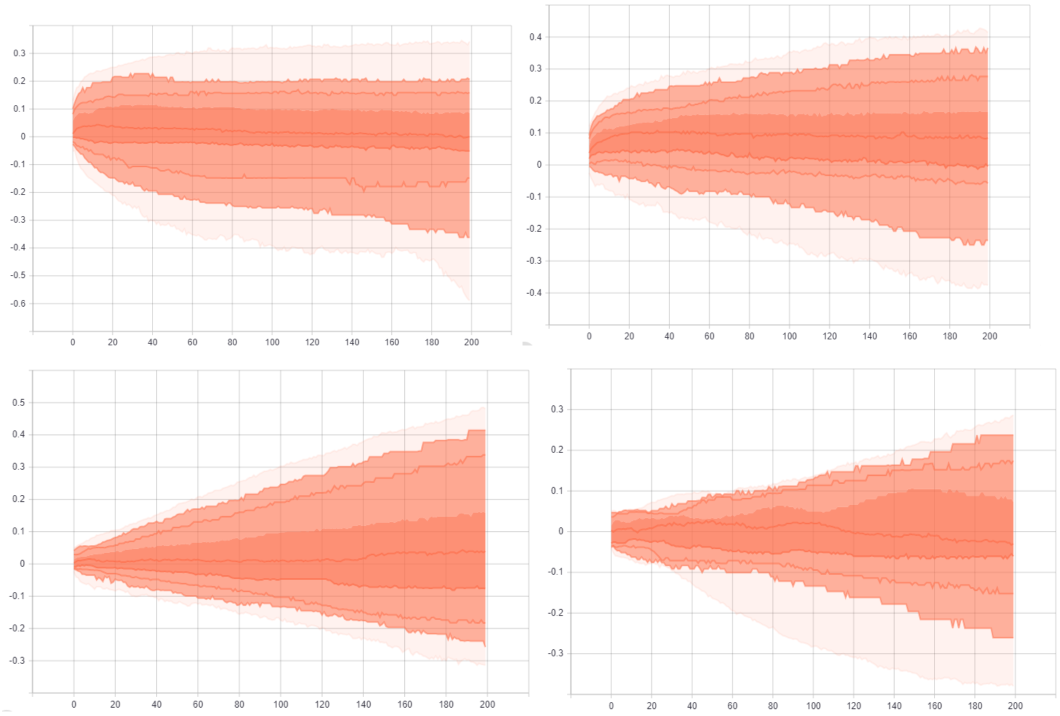
Wrist data set Distribution of bias weights for the four layers of the network using ReLu at 200 epochs with batch size 128. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



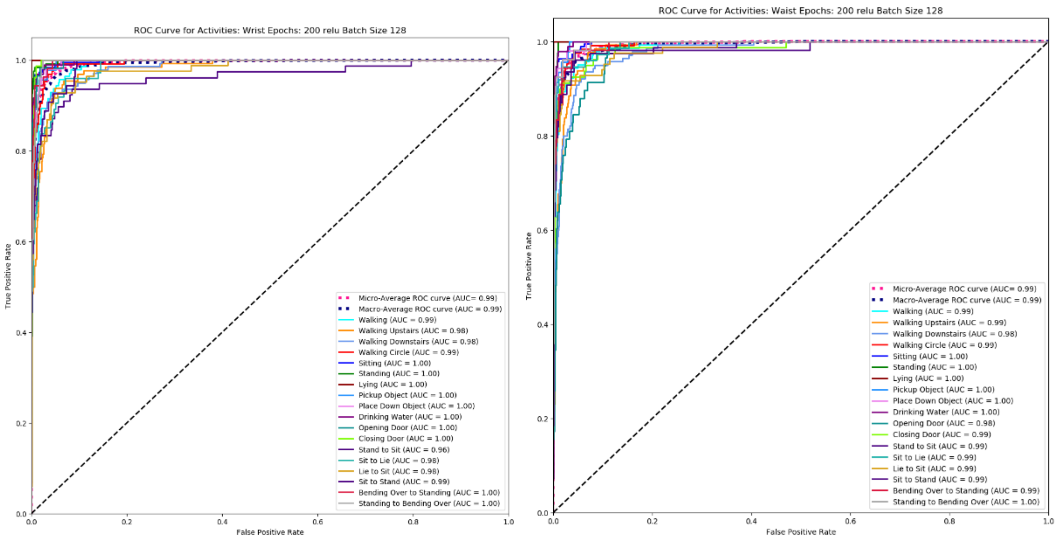
Waist data set Distribution of kernel weights for the four layers of the network using ReLu at 200 epochs with batch size 128. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



Waist data set Distribution of bias weights for the four layers of the network using ReLu at 200 epochs with batch size 128. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.

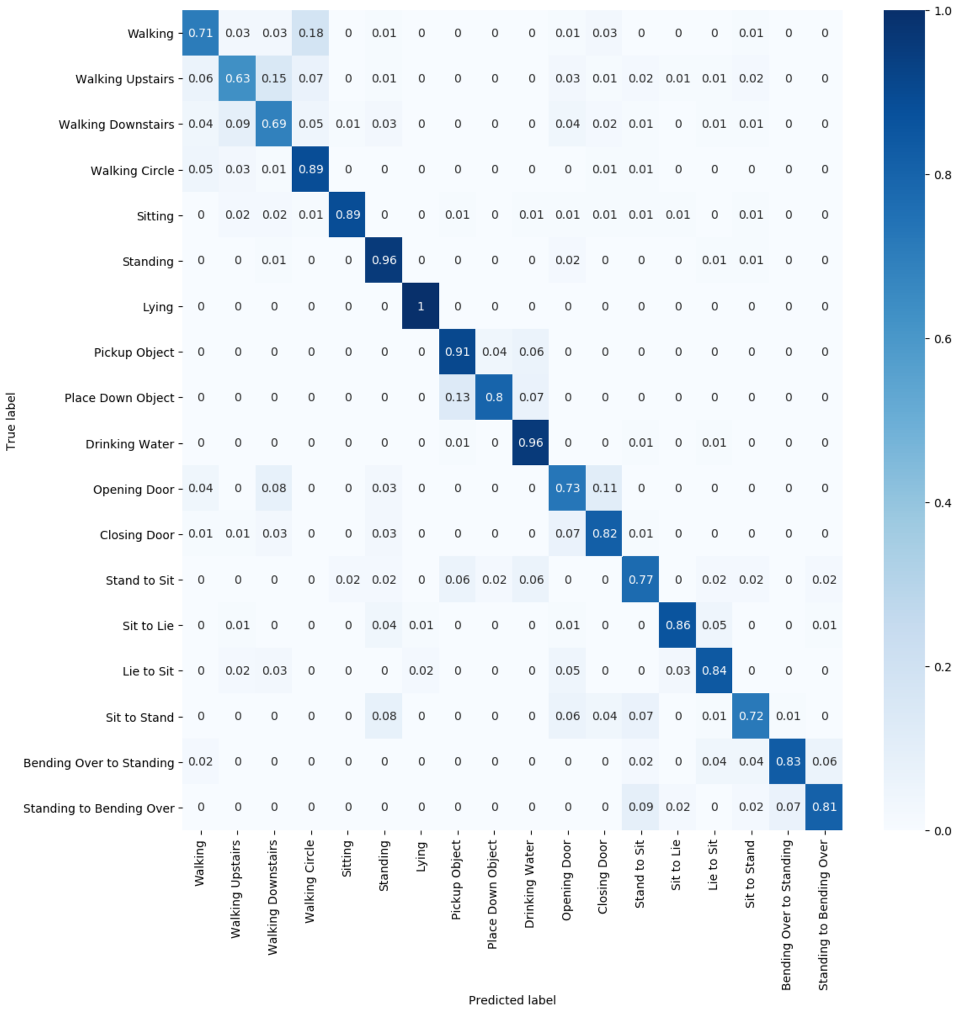


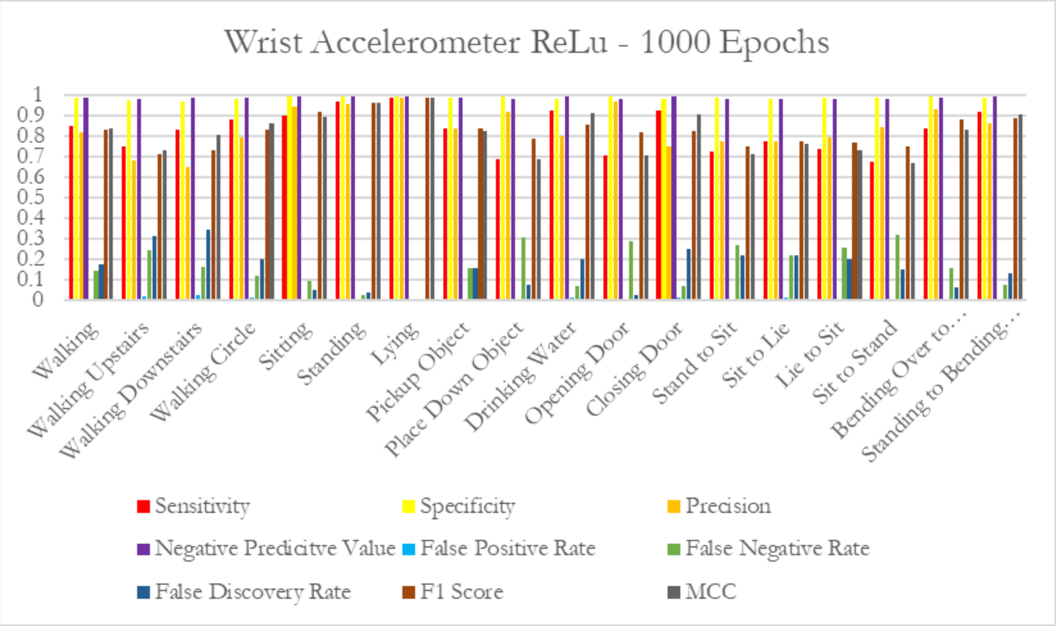
ROC Curve for wrist and waist

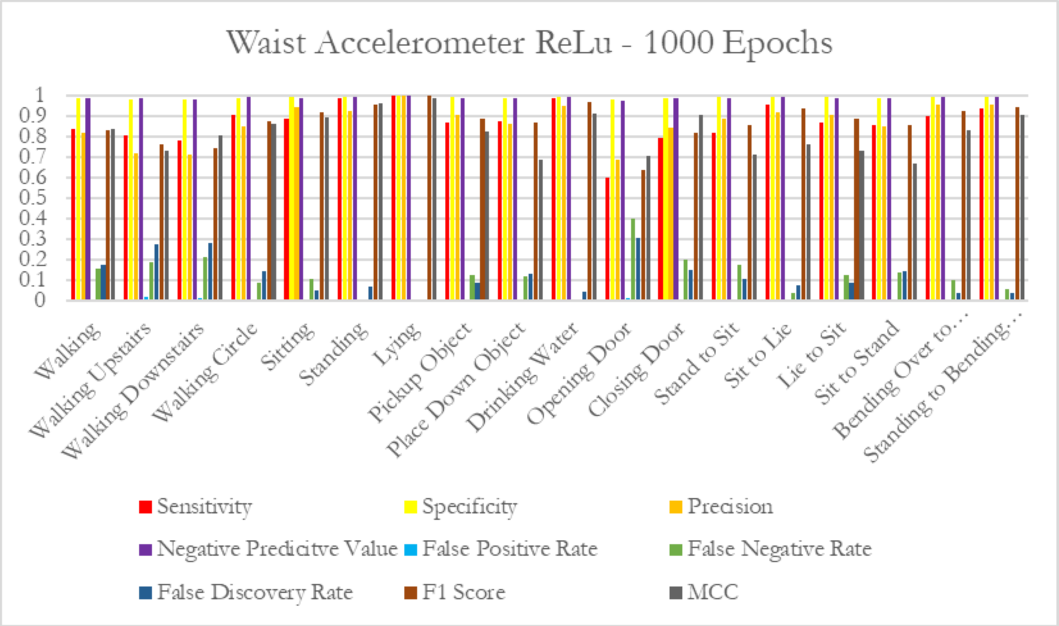


Appendix H – Additional Graphs: ReLu 1000 Epochs

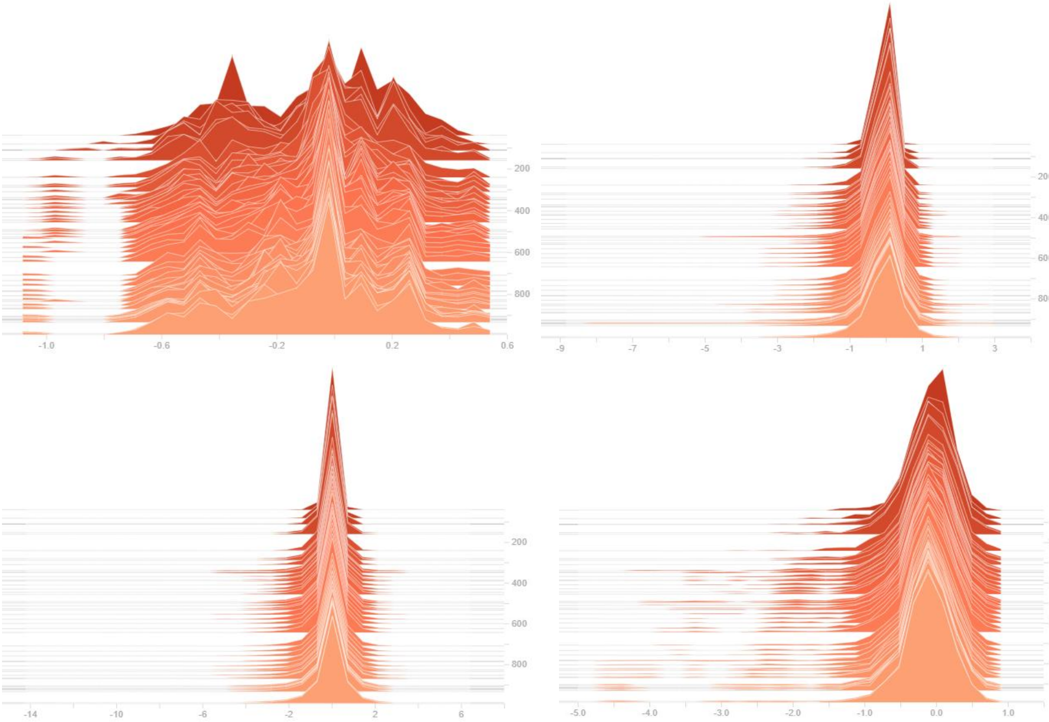
Waist confusion matrix



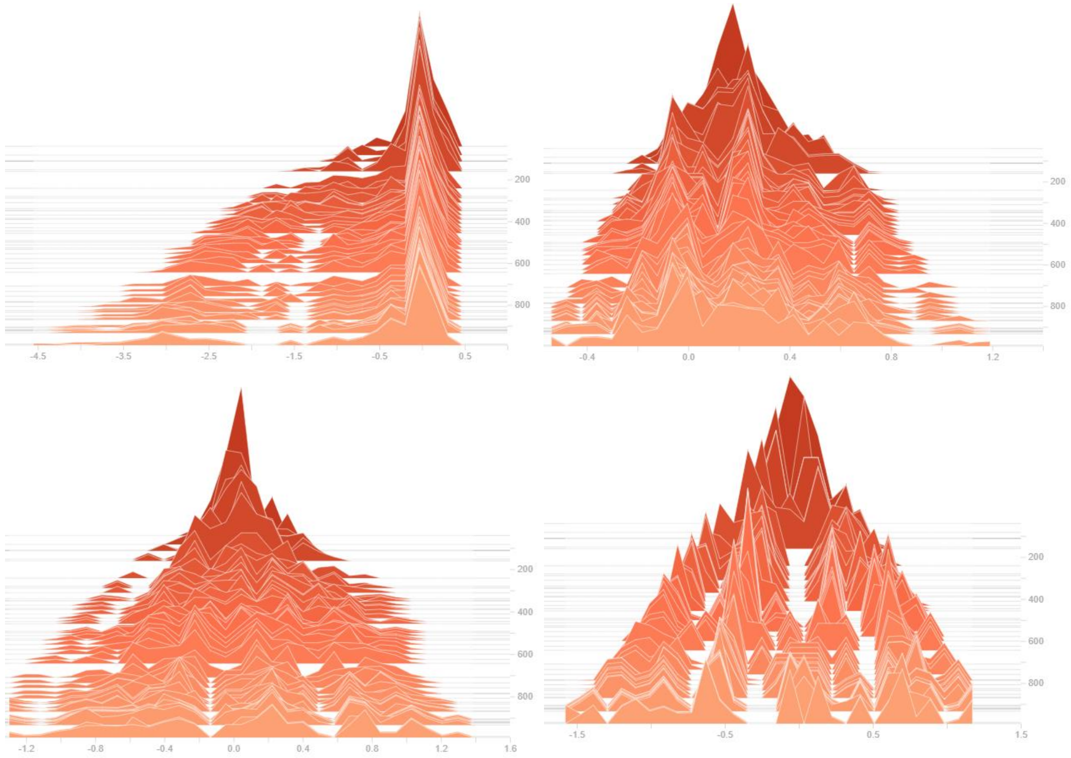




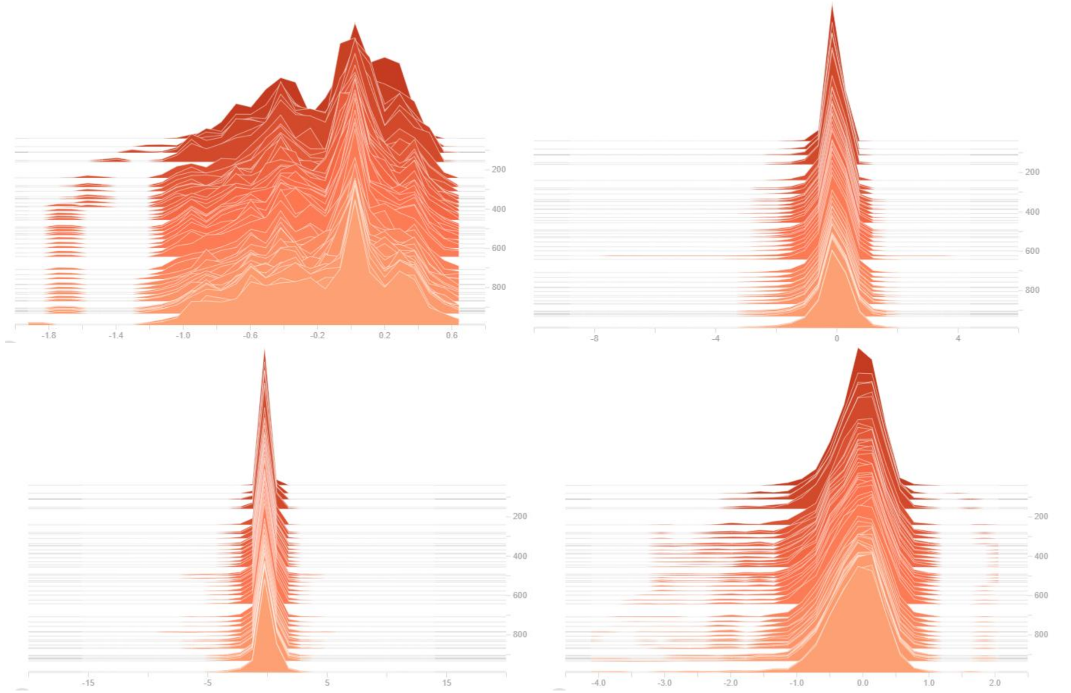
Wrist data set Histogram of kernel weights for the four layers of the network using ReLu at 1000 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



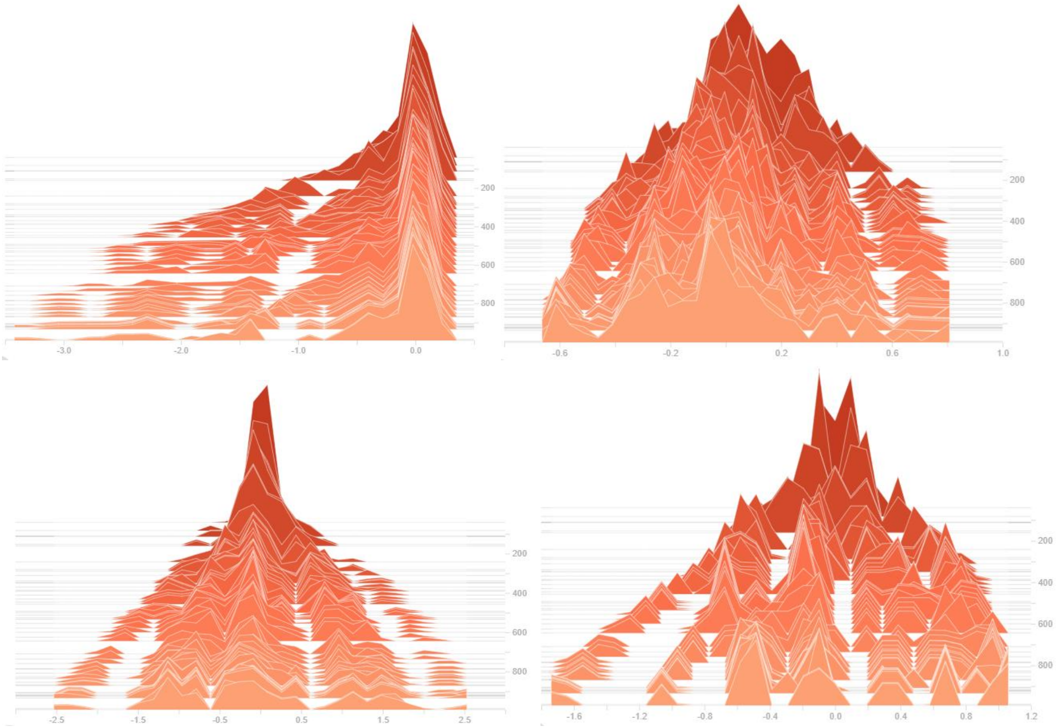
Wrist data set Histogram of bias weights for the four layers of the network using ReLu at 1000 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



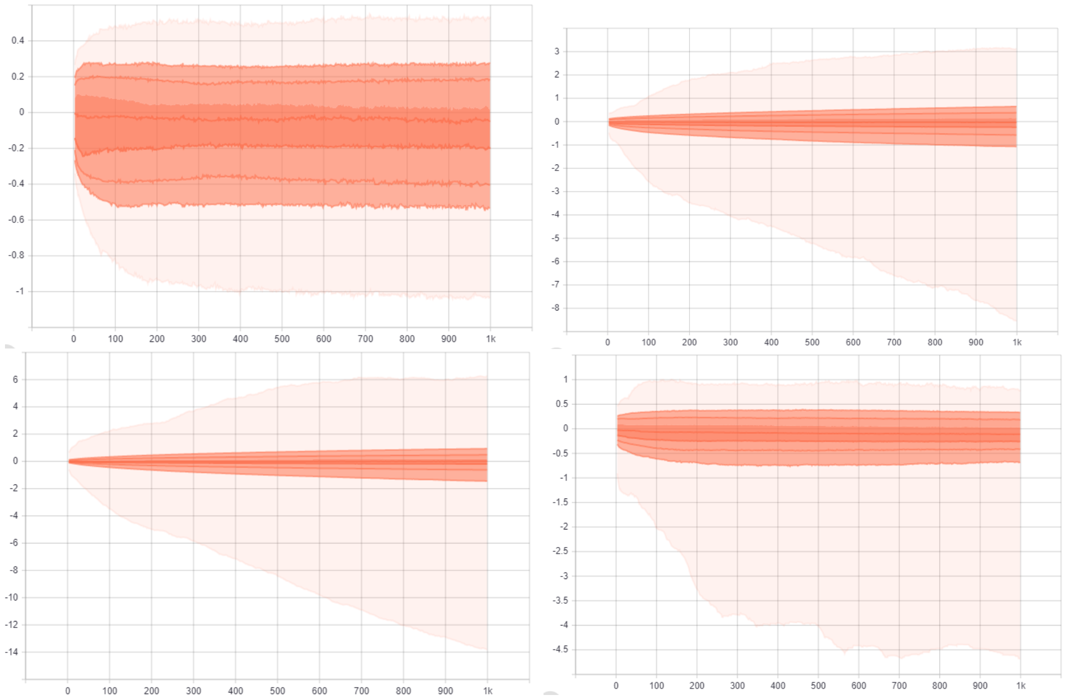
Waist data set Histogram of kernel weights for the four layers of the network using ReLu at 1000 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



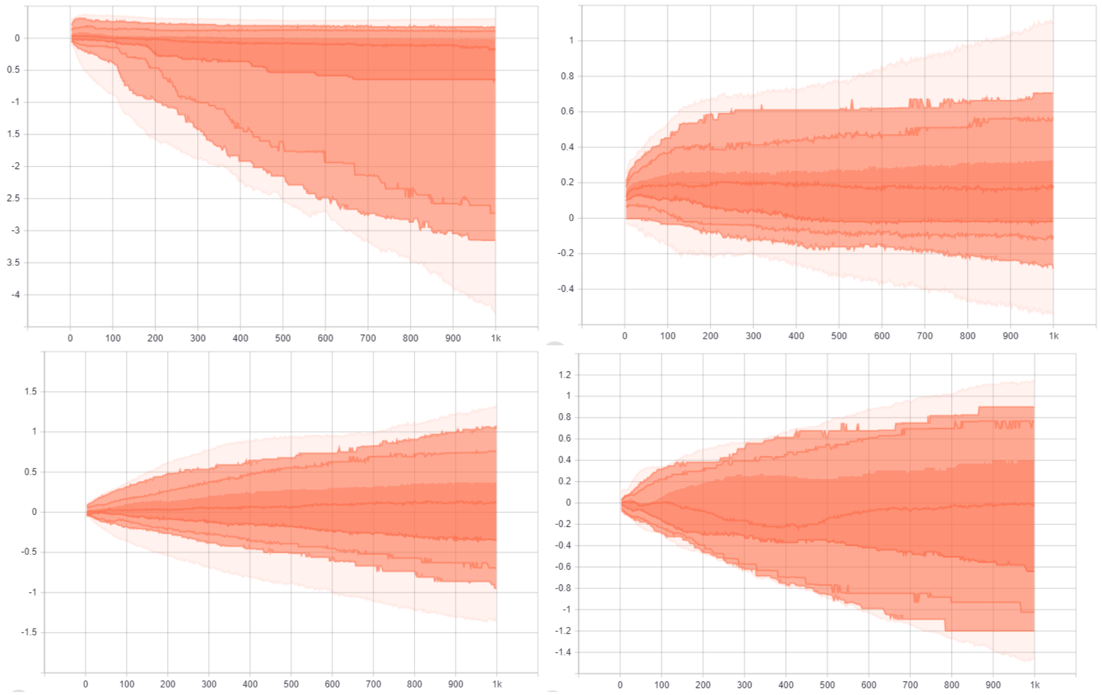
Waist data set Histogram of bias weights for the four layers of the network using ReLu at 1000 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



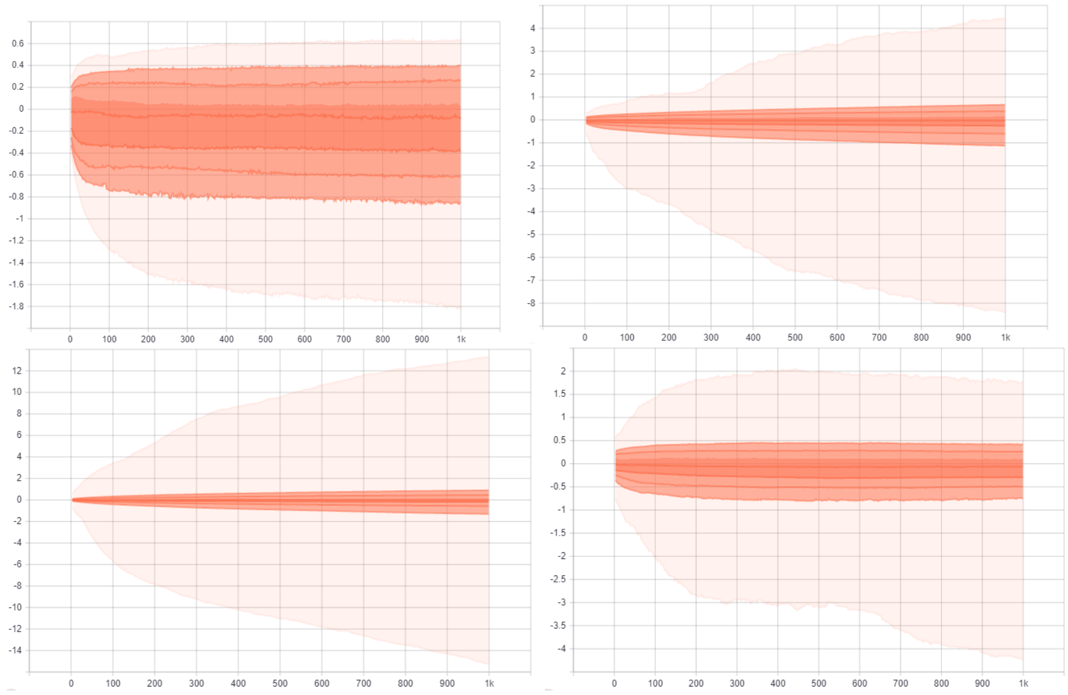
Wrist data set Distribution of kernel weights for the four layers of the network using ReLu at 1000 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



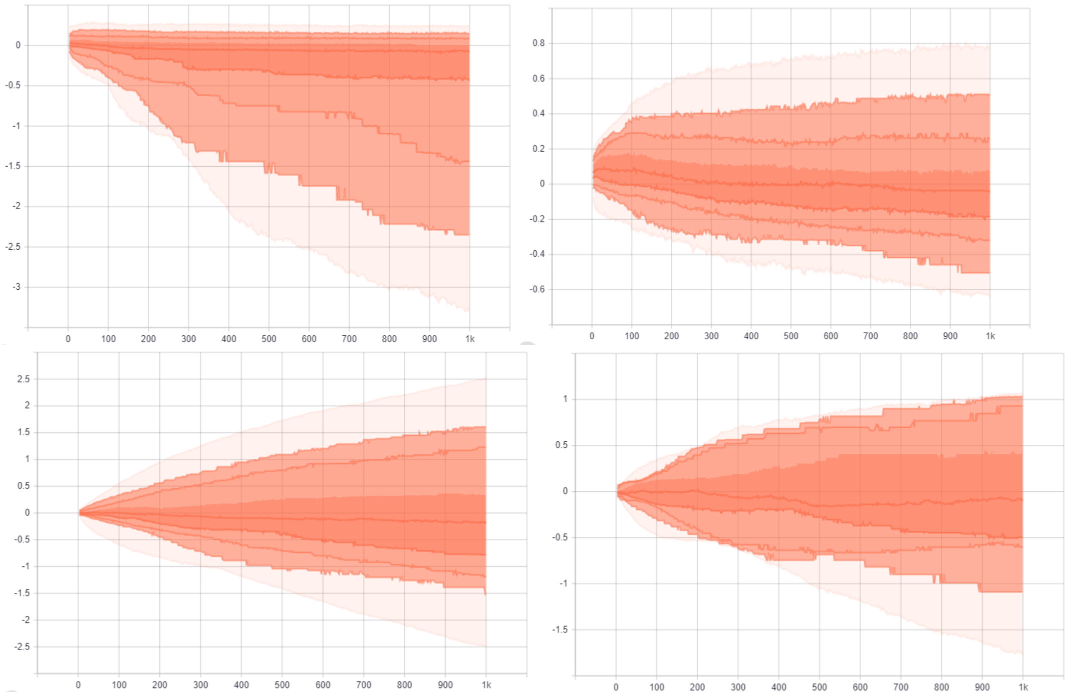
Wrist data set Distribution of bias weights for the four layers of the network using ReLu at 1000 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



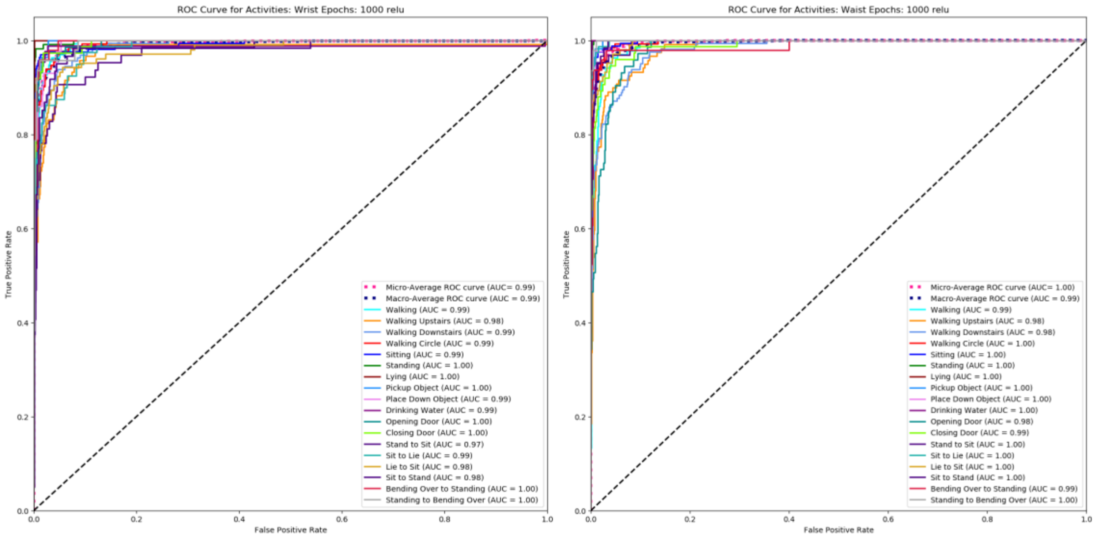
Waist data set Distribution of kernel weights for the four layers of the network using ReLu at 1000 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



Waist data set Distribution of bias weights for the four layers of the network using ReLu at 1000 epochs. From top-1left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.

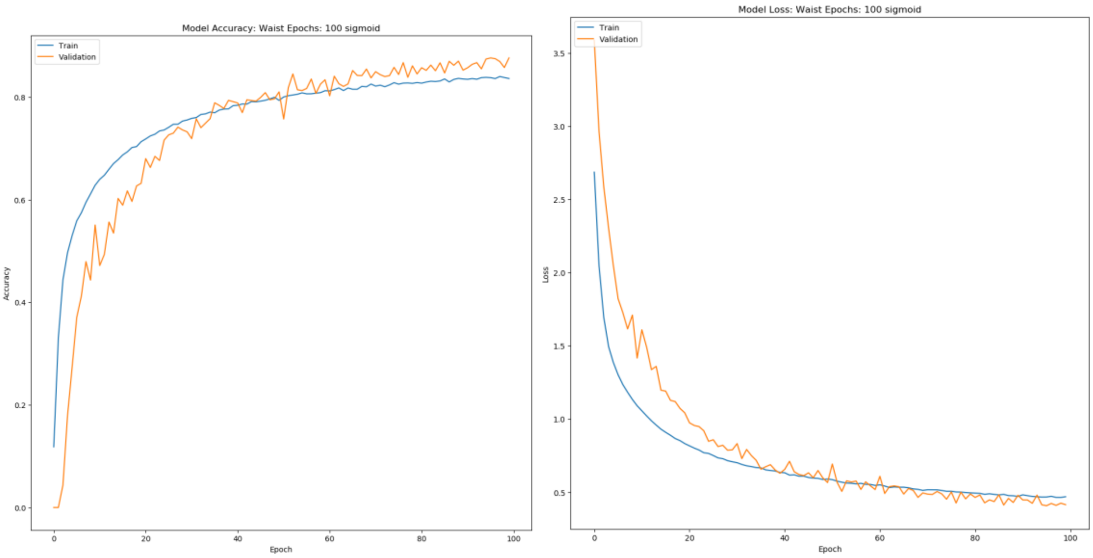


ROC Curve for wrist and waist

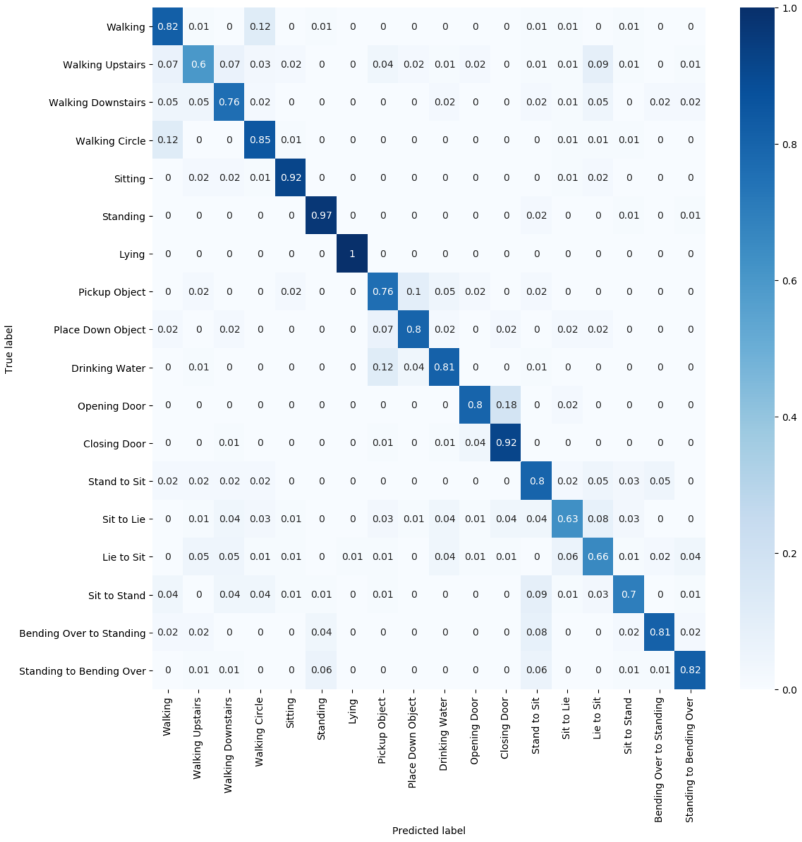


Appendix I – Additional Graphs: Sigmoid 100 Epochs

Waist Accuracy and Loss

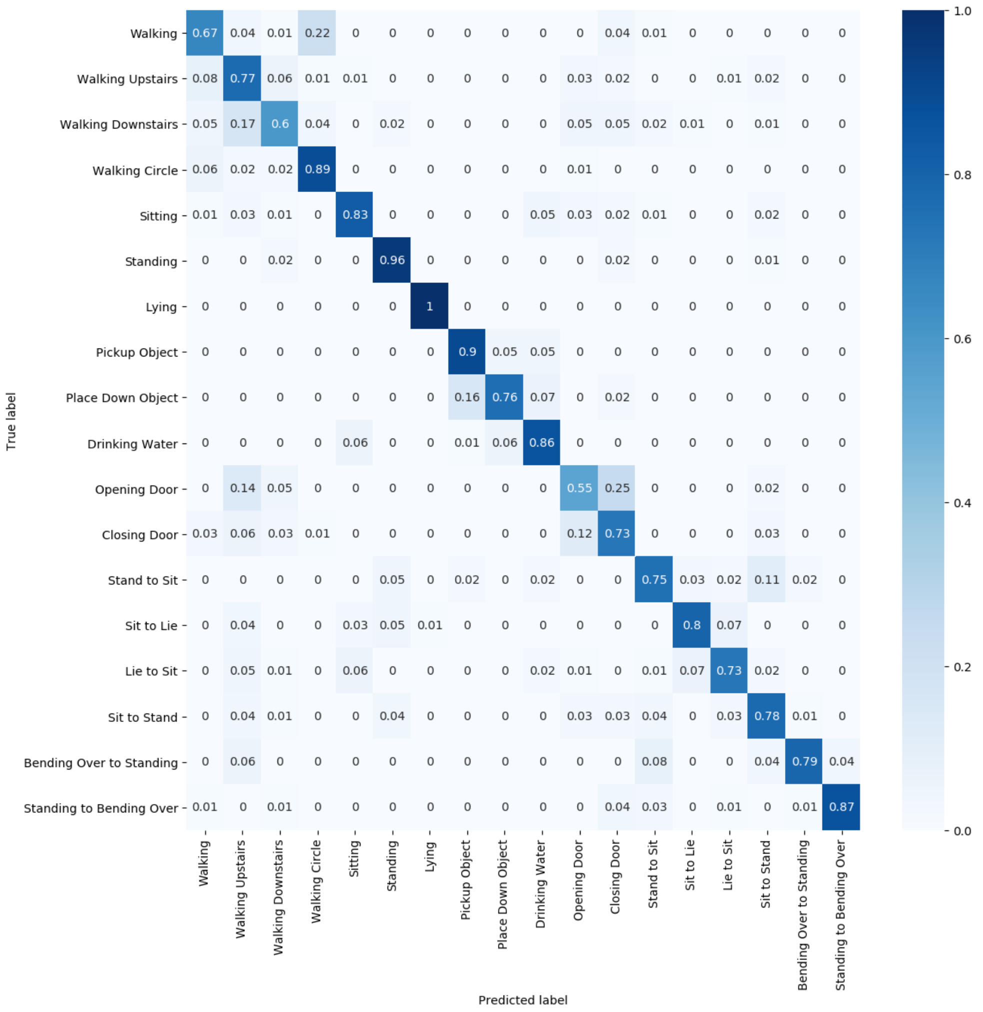


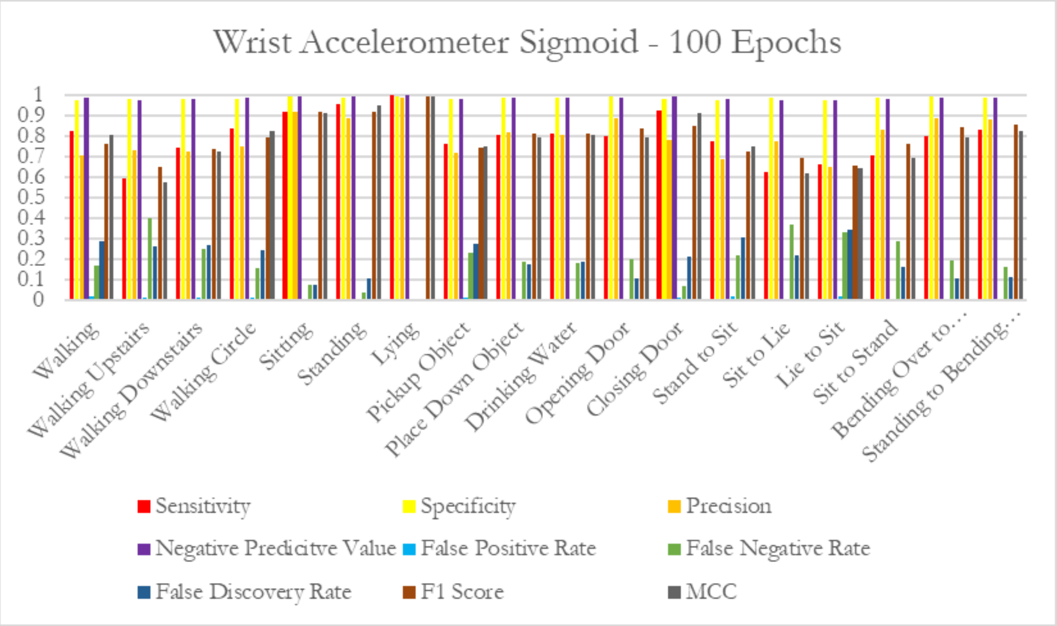
Wrist Confusion Matrix

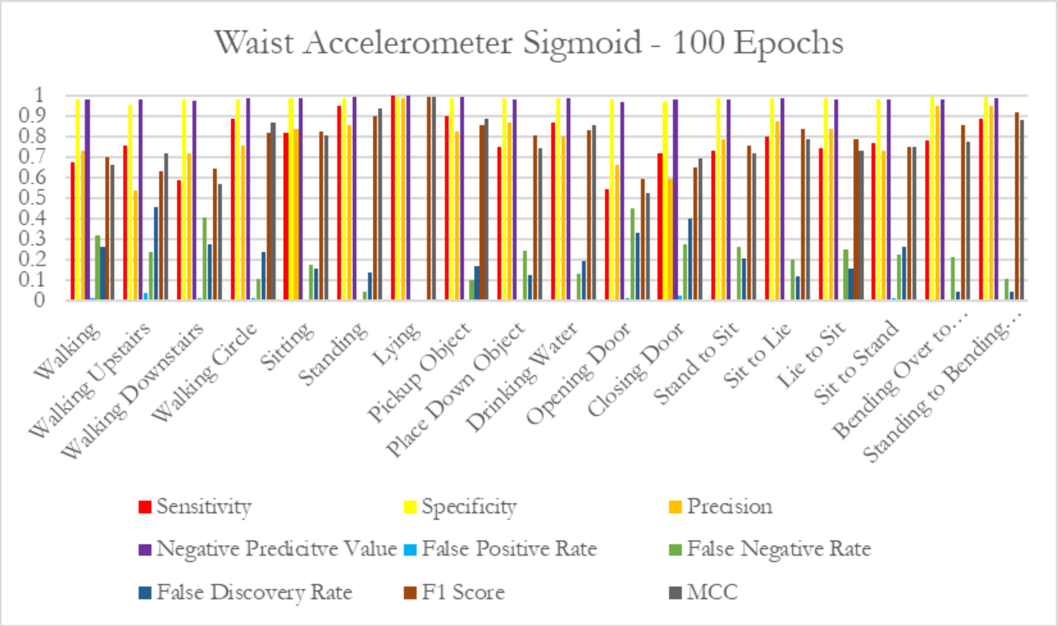


	Walking	Walking Upstairs	Walking Downstairs	Walking Circle	Sitting	Standing	Lying	Pickup Object	Place Down Object	Drinking Water	Opening Door	Closing Door	Stand to Sit	Sit to Lie	Lie to Sit	Sit to Stand	Bending Over to Standing	Standing to Bending Over
Walking	0.82	0.01	0	0.12	0	0.01	0	0	0	0	0	0	0.01	0.01	0	0.01	0	0
Walking Upstairs	0.07	0.6	0.07	0.03	0.02	0	0	0.04	0.02	0.01	0.02	0	0.01	0.01	0.09	0.01	0	0.01
Walking Downstairs	0.05	0.05	0.76	0.02	0	0	0	0	0	0.02	0	0	0.02	0.01	0.05	0	0.02	0.02
Walking Circle	0.12	0	0	0.85	0.01	0	0	0	0	0	0	0	0	0.01	0.01	0.01	0	0
Sitting	0	0.02	0.02	0.01	0.92	0	0	0	0	0	0	0	0	0.01	0.02	0	0	0
Standing	0	0	0	0	0	0.97	0	0	0	0	0	0	0.02	0	0	0.01	0	0.01
Lying	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Pickup Object	0	0.02	0	0	0.02	0	0	0.76	0.1	0.05	0.02	0	0.02	0	0	0	0	0
Place Down Object	0.02	0	0.02	0	0	0	0	0.07	0.8	0.02	0	0.02	0	0.02	0.02	0	0	0
Drinking Water	0	0.01	0	0	0	0	0	0.12	0.04	0.81	0	0	0.01	0	0	0	0	0
Opening Door	0	0	0	0	0	0	0	0	0	0	0.8	0.18	0	0.02	0	0	0	0
Closing Door	0	0	0.01	0	0	0	0	0.01	0	0.01	0.04	0.92	0	0	0	0	0	0
Stand to Sit	0.02	0.02	0.02	0.02	0	0	0	0	0	0	0	0	0.8	0.02	0.05	0.03	0.05	0
Sit to Lie	0	0.01	0.04	0.03	0.01	0	0	0.03	0.01	0.04	0.01	0.04	0.04	0.63	0.08	0.03	0	0
Lie to Sit	0	0.05	0.05	0.01	0.01	0	0.01	0.01	0	0.04	0.01	0.01	0	0.06	0.66	0.01	0.02	0.04
Sit to Stand	0.04	0	0.04	0.04	0.01	0.01	0	0.01	0	0	0	0	0.09	0.01	0.03	0.7	0	0.01
Bending Over to Standing	0.02	0.02	0	0	0	0.04	0	0	0	0	0	0	0.08	0	0	0.02	0.81	0.02
Standing to Bending Over	0	0.01	0.01	0	0	0.06	0	0	0	0	0	0	0.06	0	0	0.01	0.01	0.82

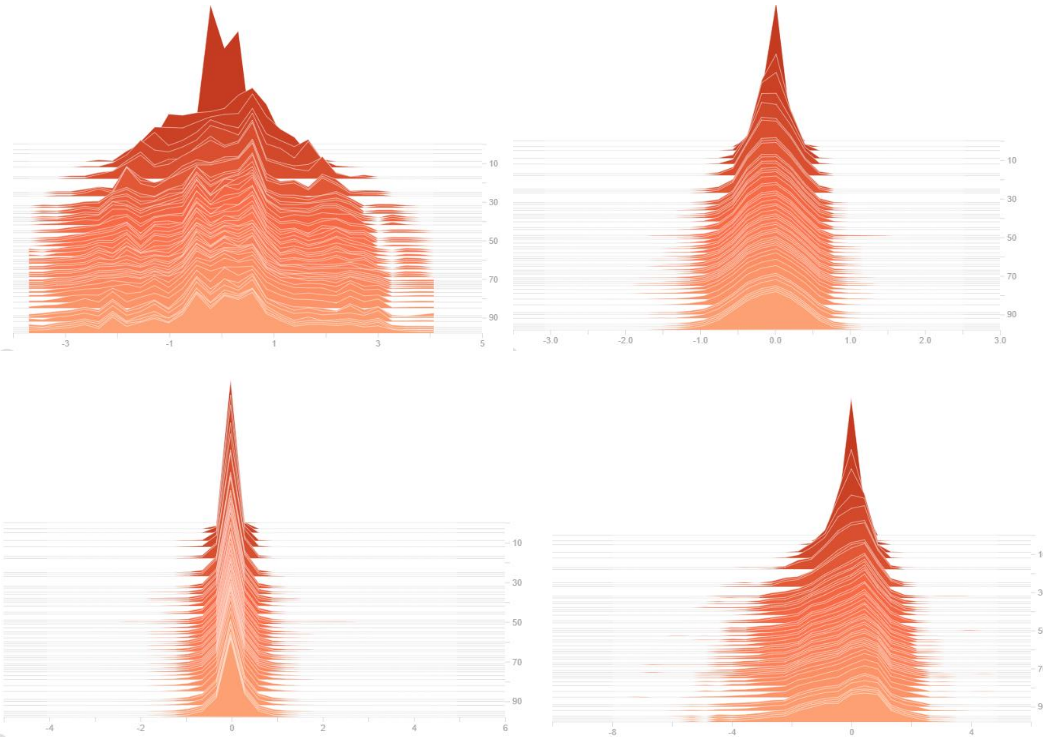
Waist Confusion Matrix



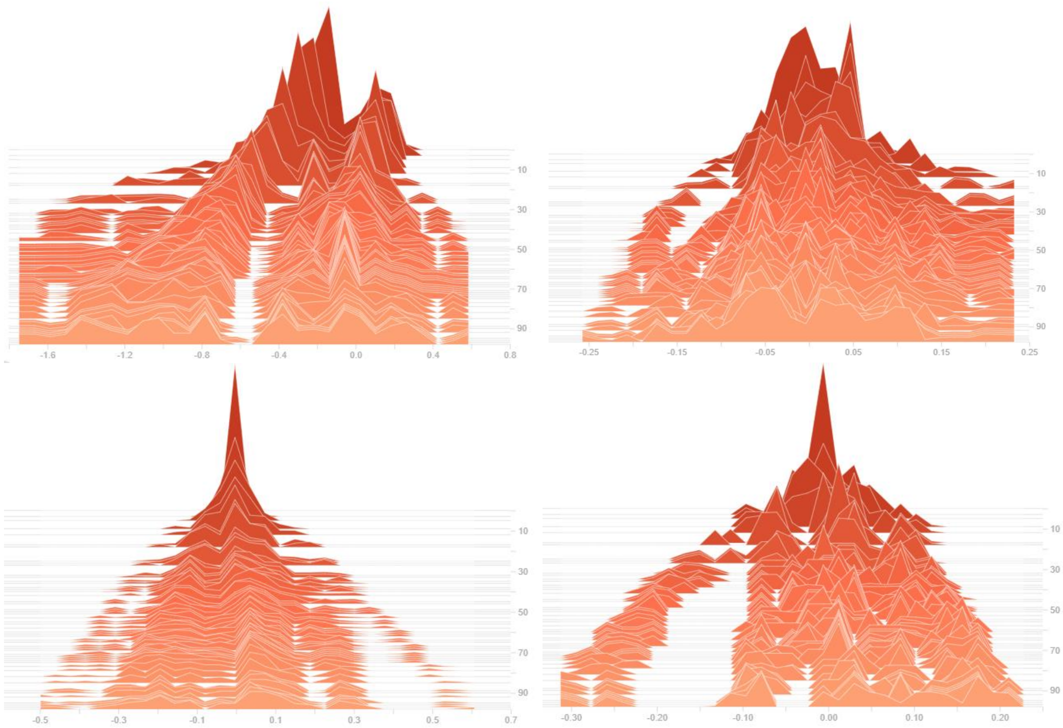




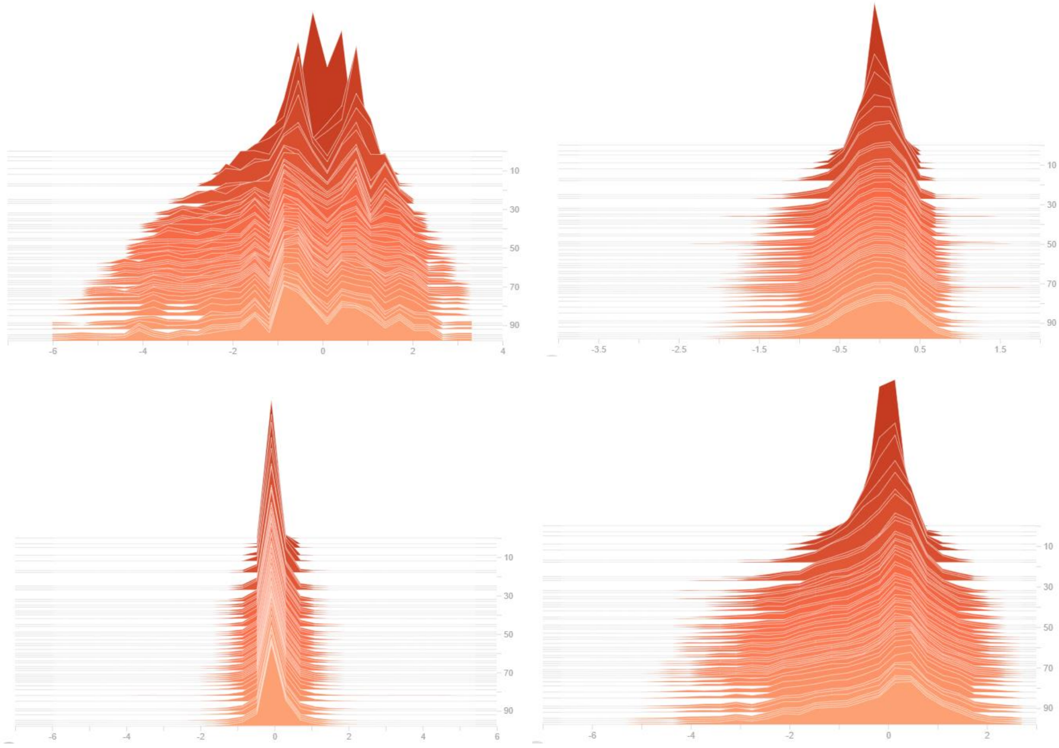
Wrist data set Histogram of kernel weights for the four layers of the network using Sigmoid at 100 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



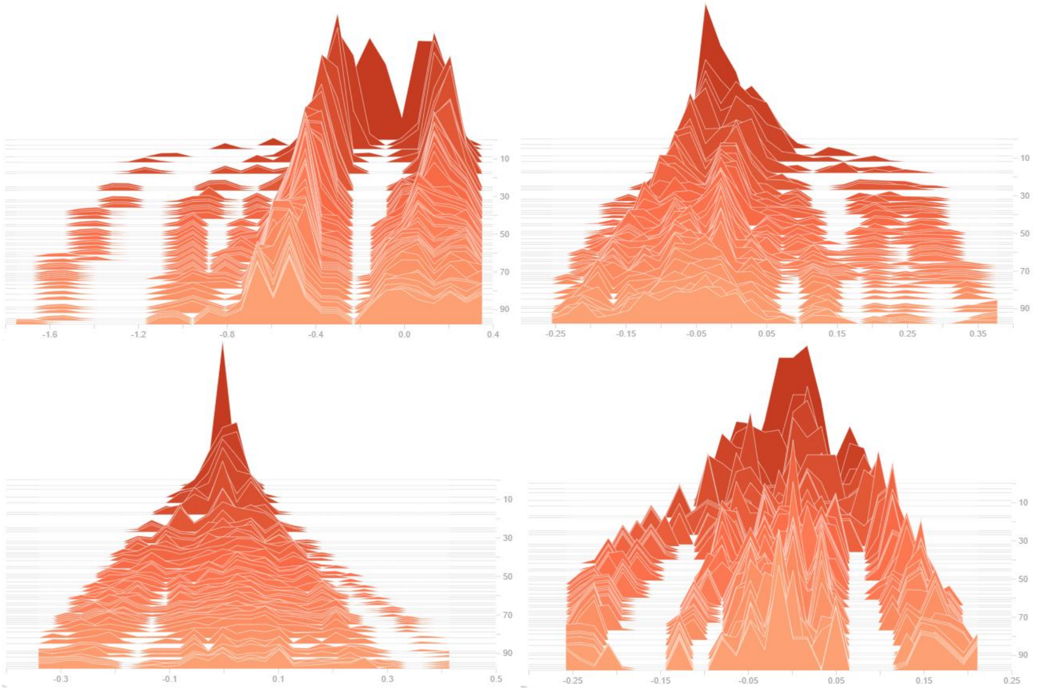
Wrist data set Histogram of bias weights for the four layers of the network using Sigmoid at 100 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



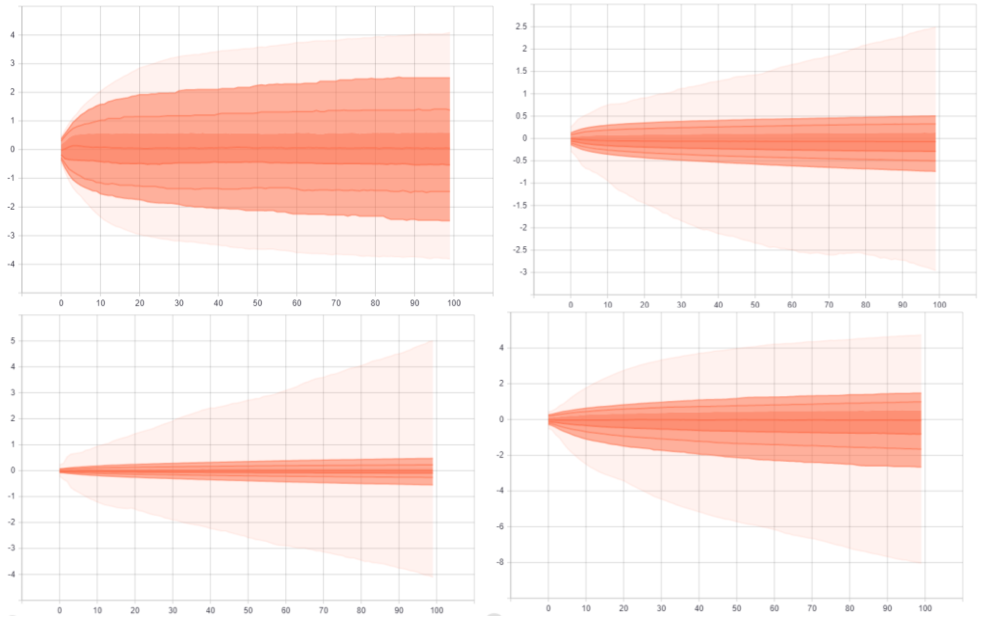
Waist data set Histogram of kernel weights for the four layers of the network using Sigmoid at 100 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



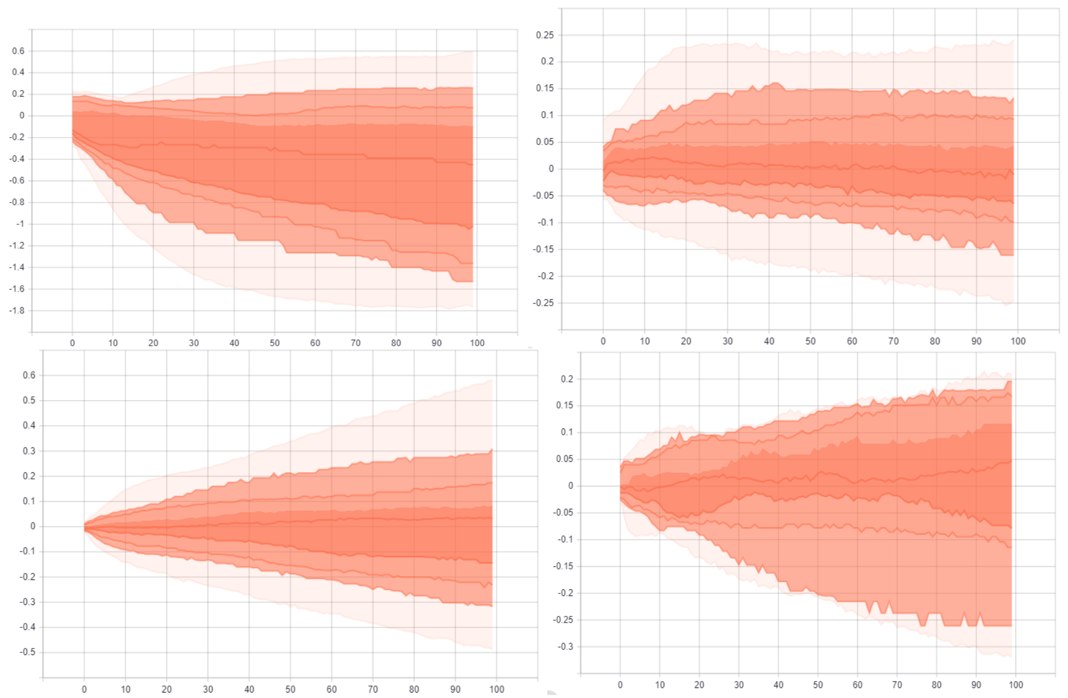
Waist data set Histogram of bias weights for the four layers of the network using Sigmoid at 100 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



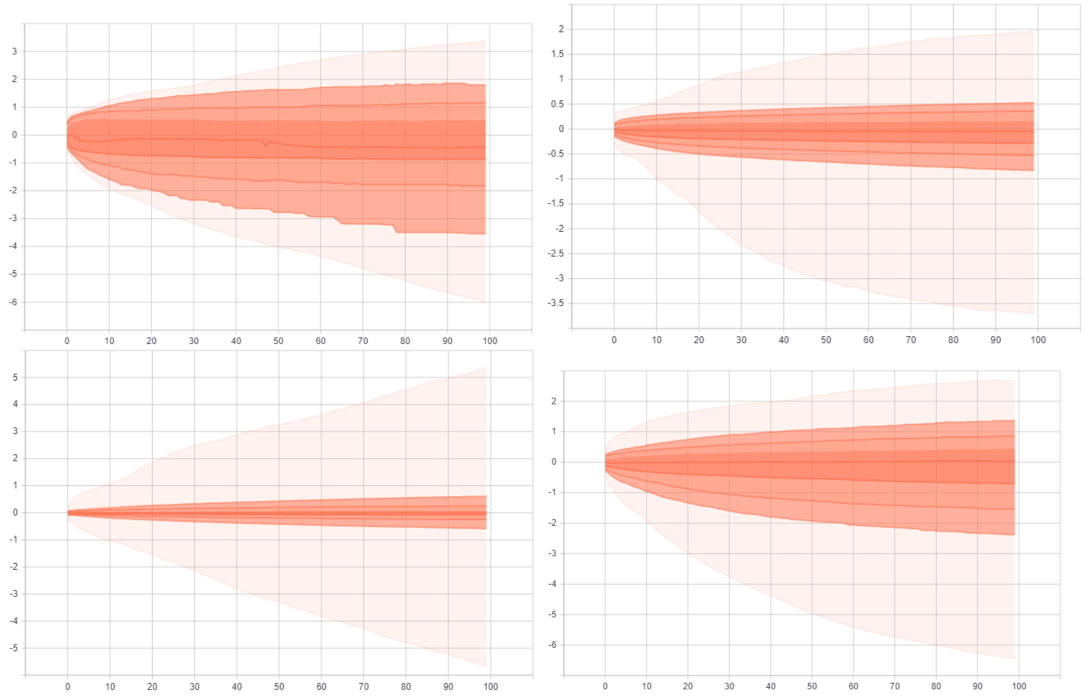
Wrist data set Distribution of kernel weights for the four layers of the network using Sigmoid at 100 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



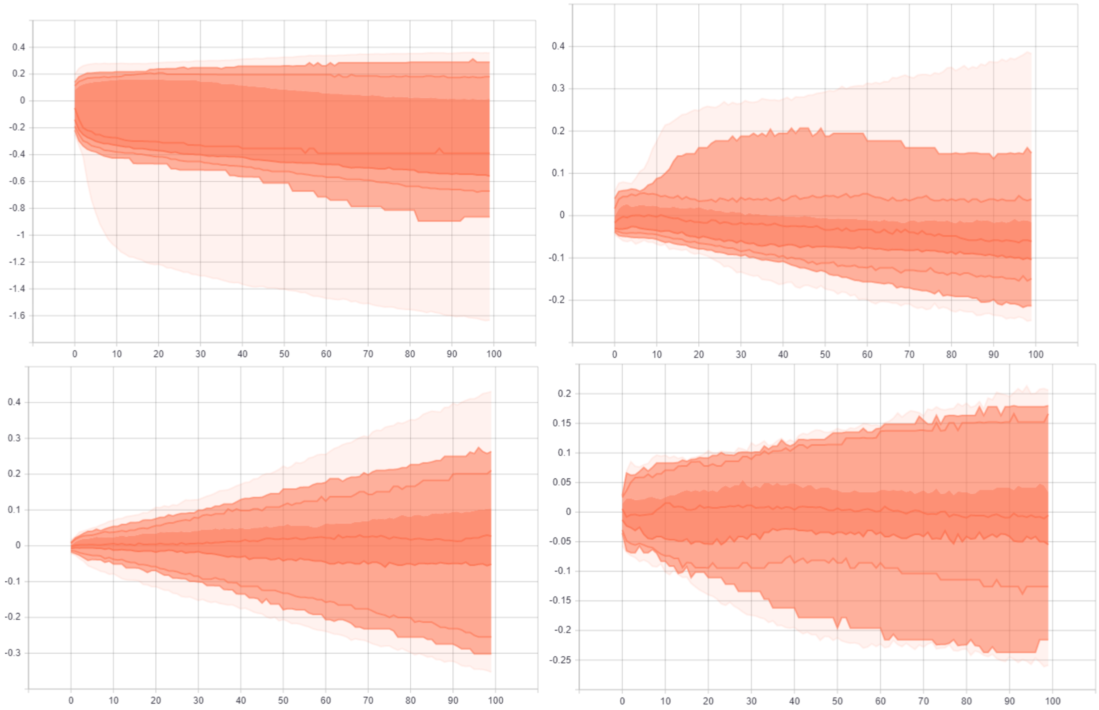
Wrist data set Distribution of bias weights for the four layers of the network using Sigmoid at 100 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



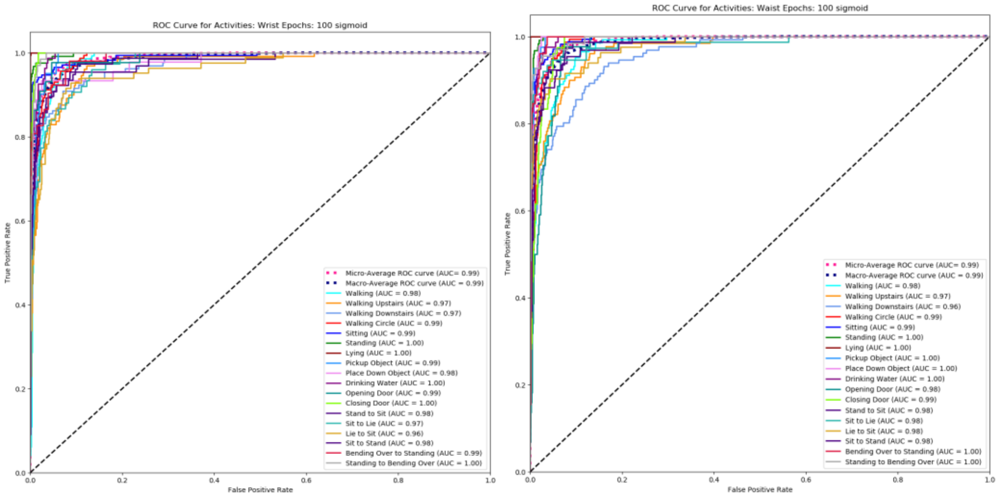
Waist data set Distribution of kernel weights for the four layers of the network using Sigmoid at 100 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



Waist data set Distribution of bias weights for the four layers of the network using Sigmoid at 100 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.

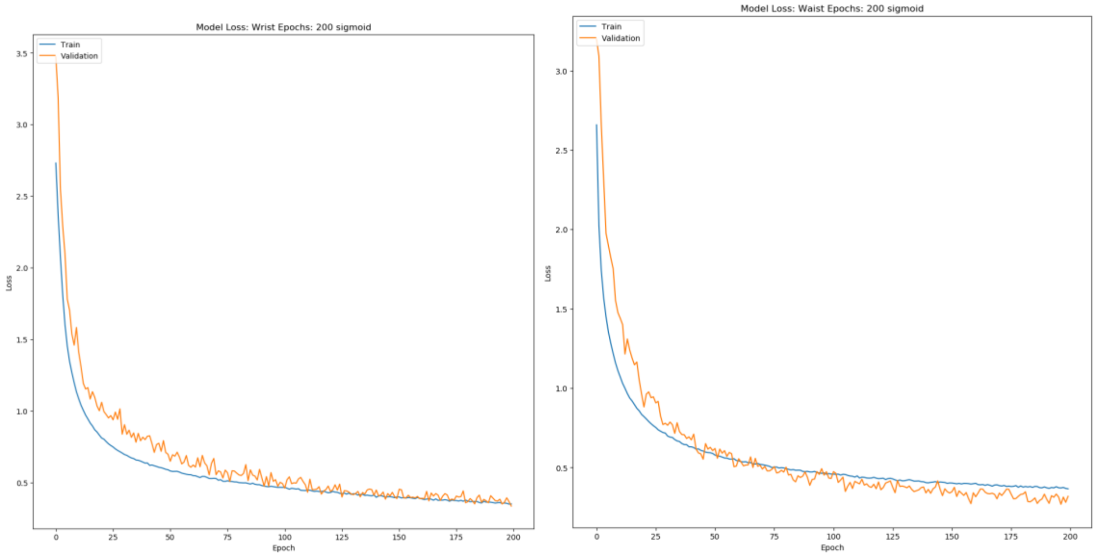


ROC Curve for wrist and waist

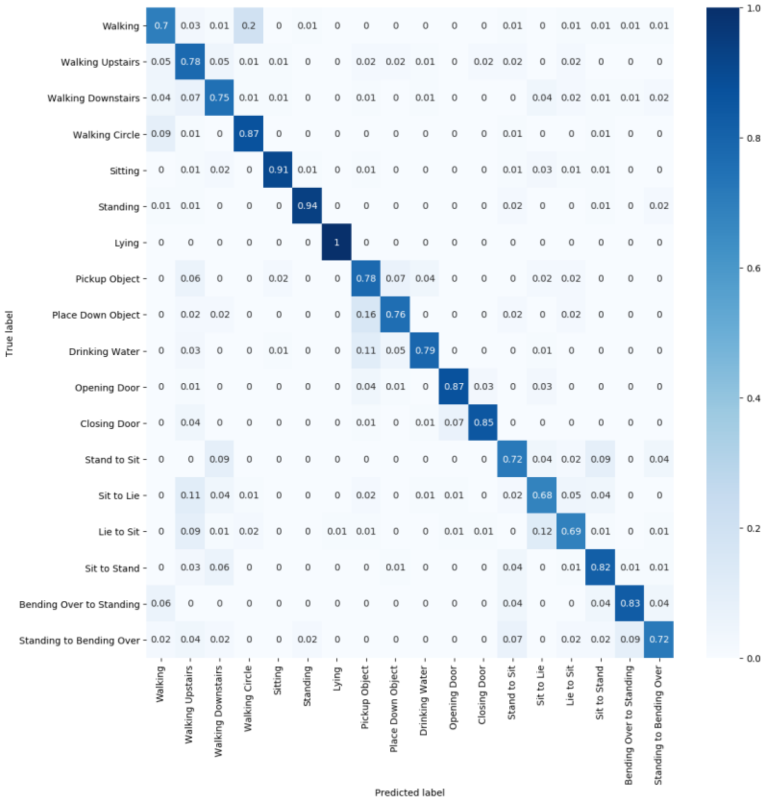


Appendix J – Additional Graphs: Sigmoid 200 Epochs

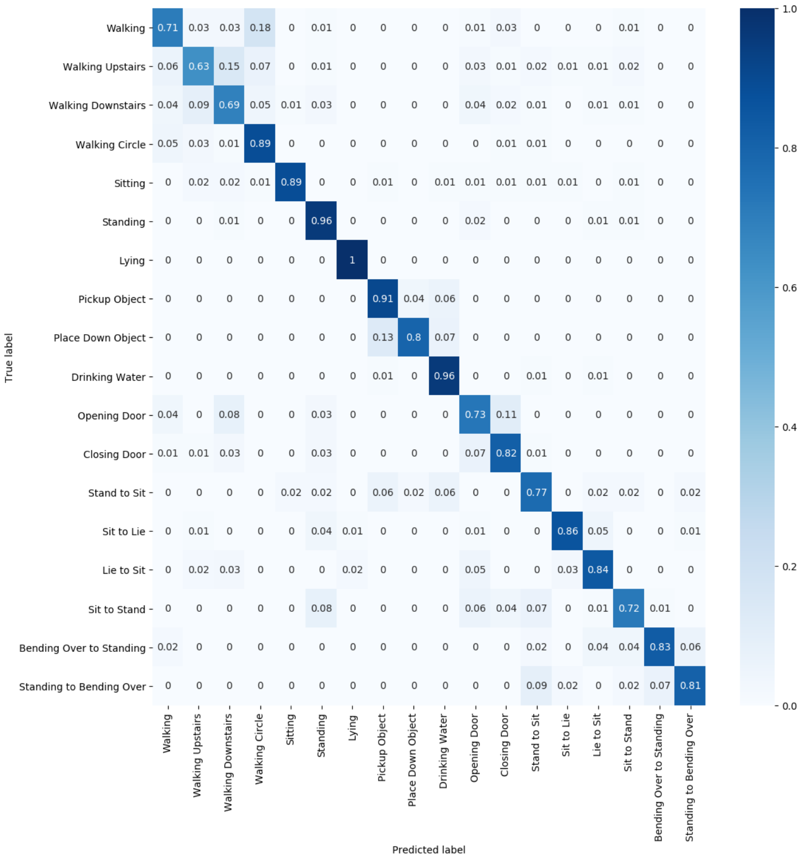
Loss of Wrist and Waist

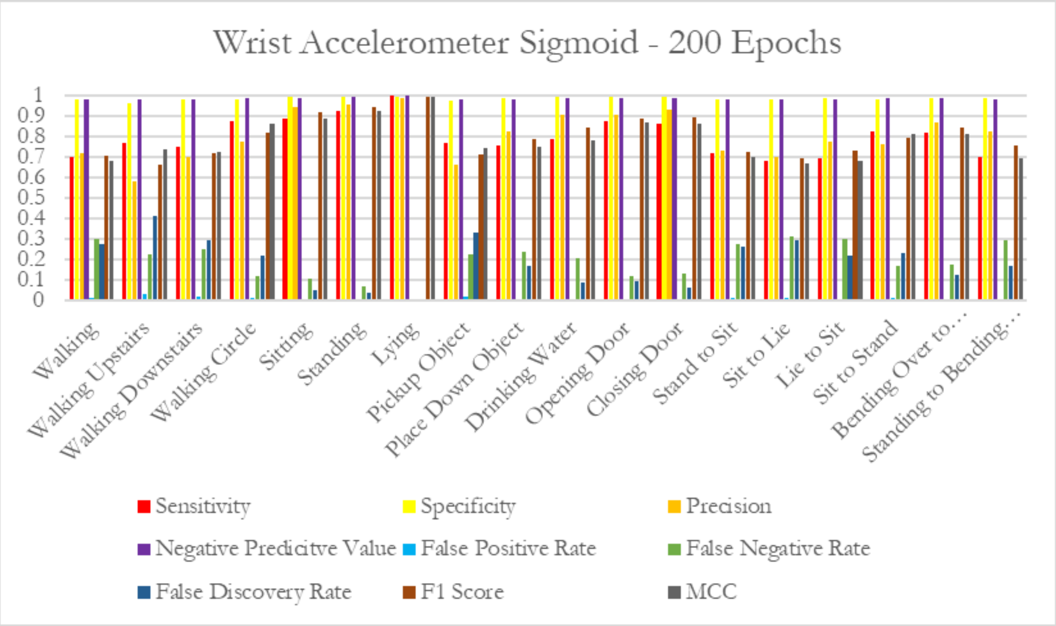


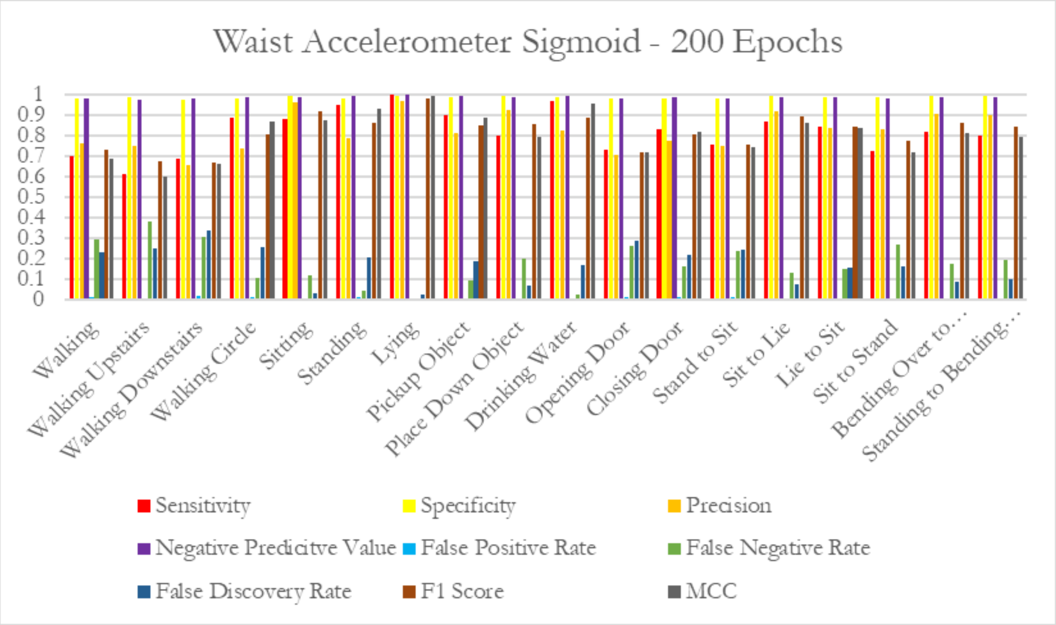
Wrist confusion matrix



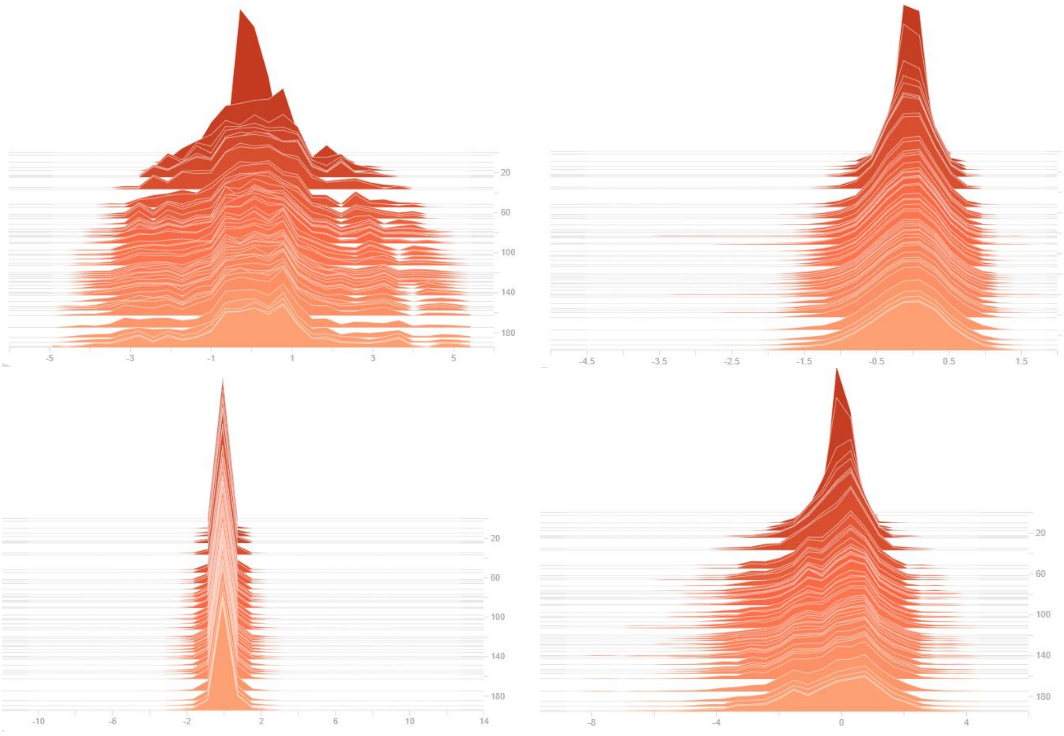
Waist confusion matrix



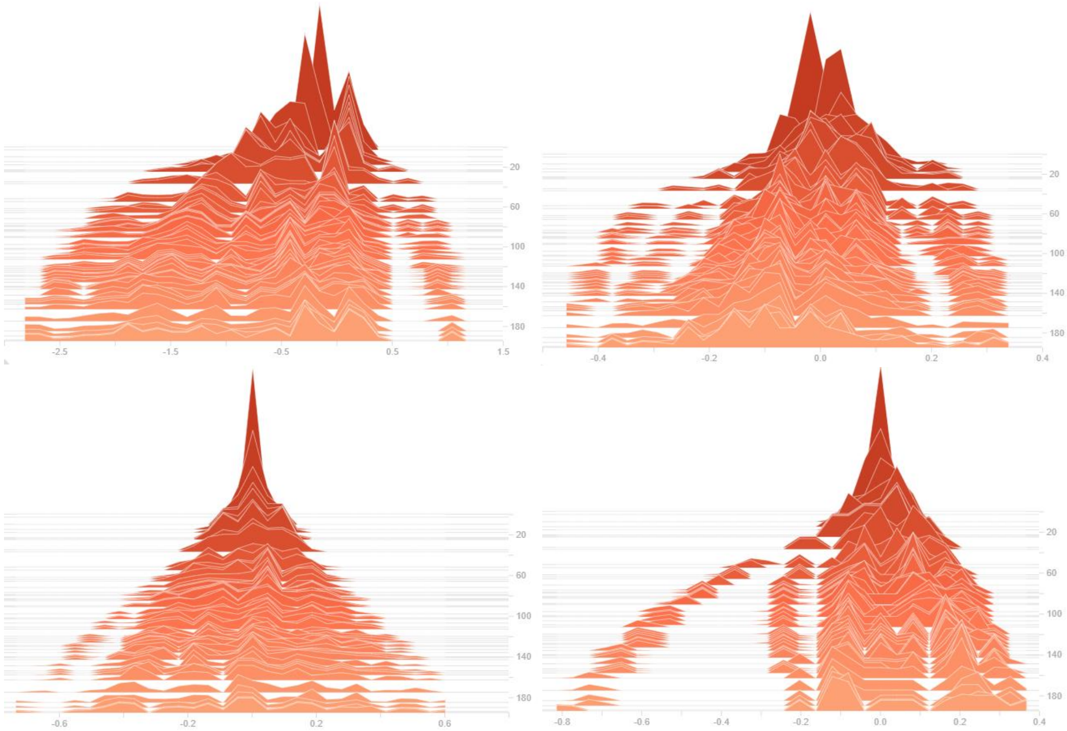




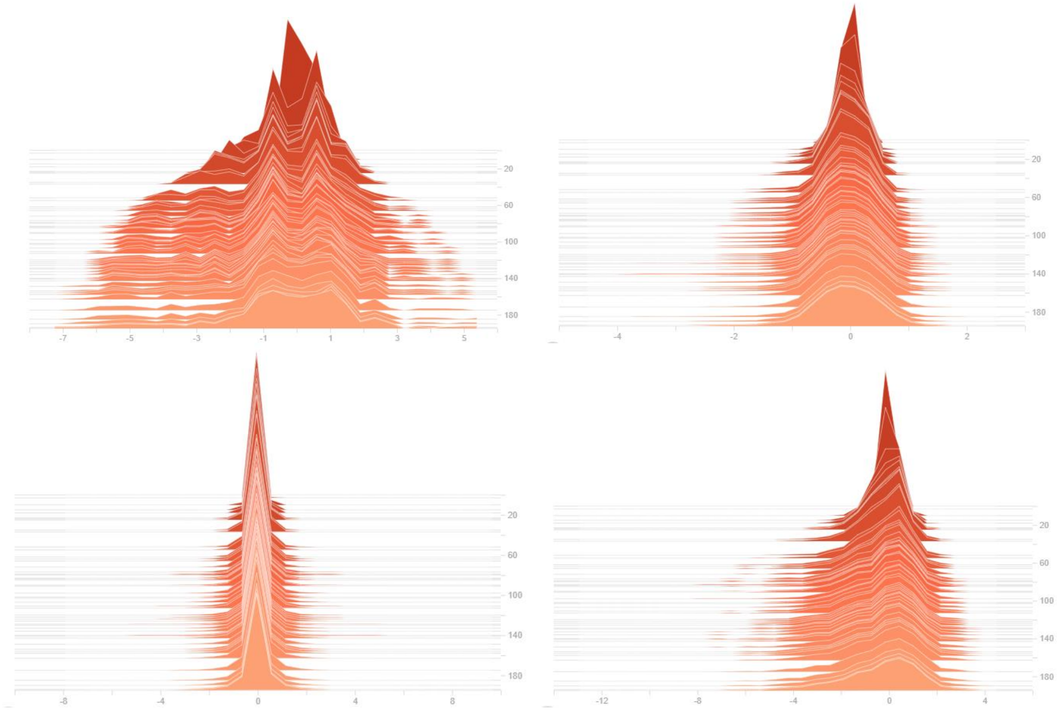
Wrist data set Histogram of kernel weights for the four layers of the network using Sigmoid at 200 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



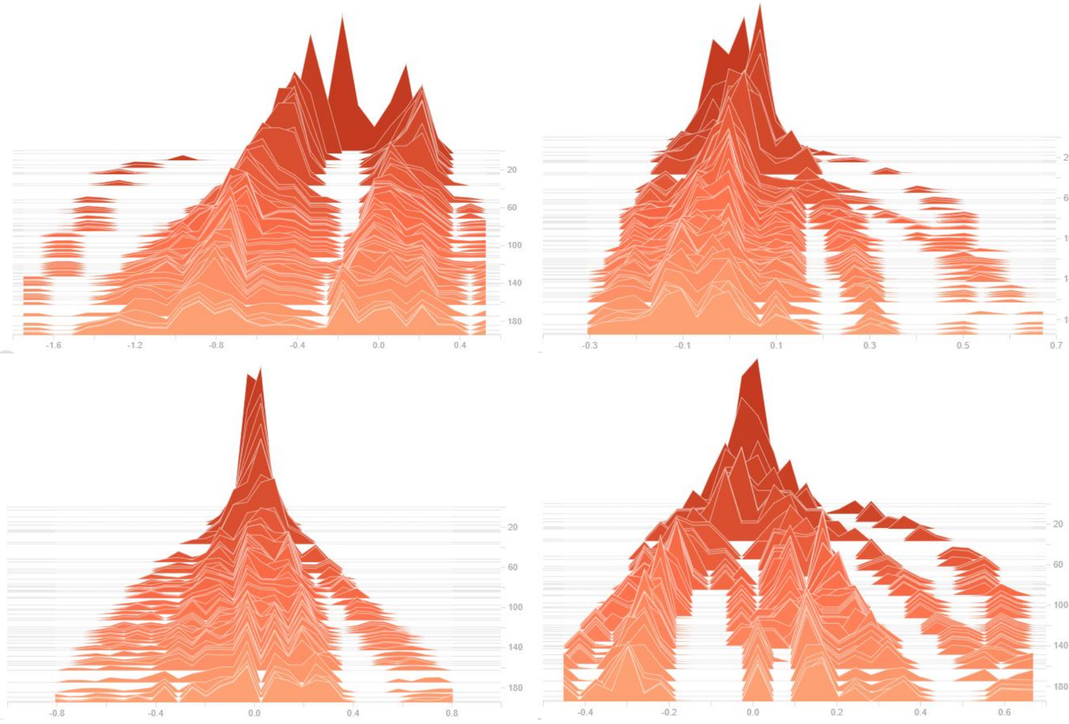
Wrist data set Histogram of bias weights for the four layers of the network using Sigmoid at 200 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



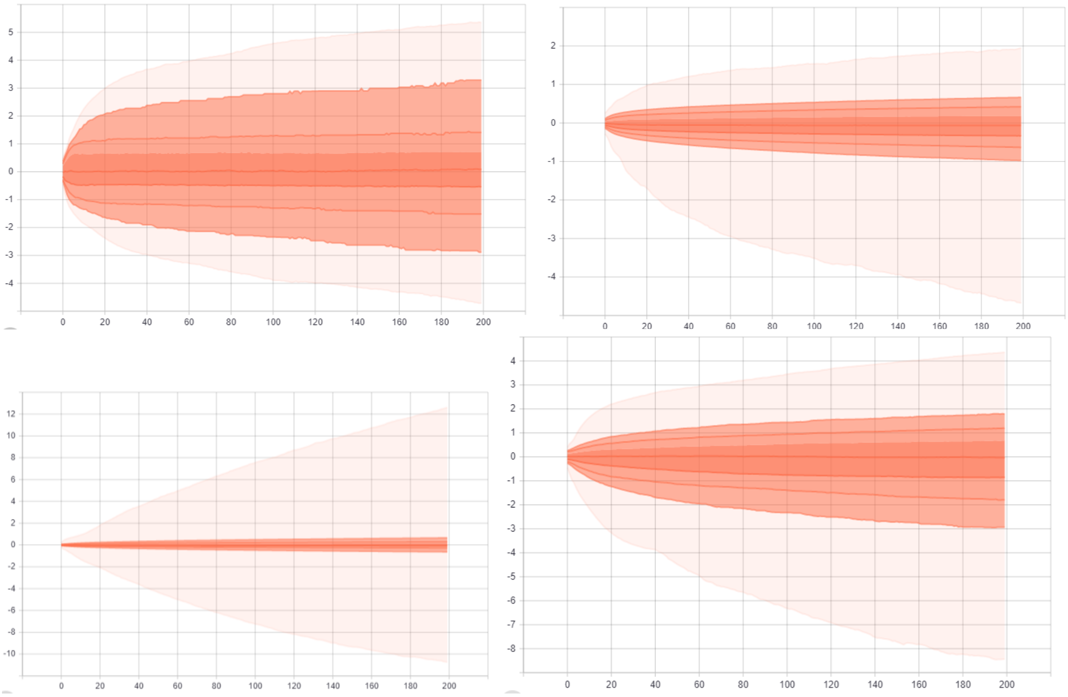
Waist data set Histogram of kernel weights for the four layers of the network using Sigmoid at 200 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



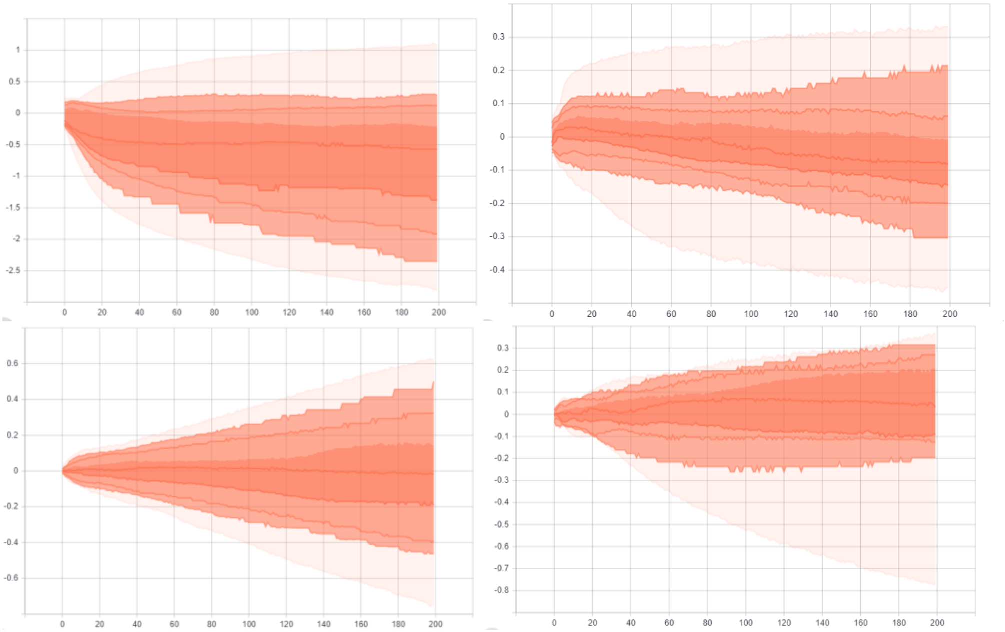
Waist data set Histogram of bias weights for the four layers of the network using Sigmoid at 200 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



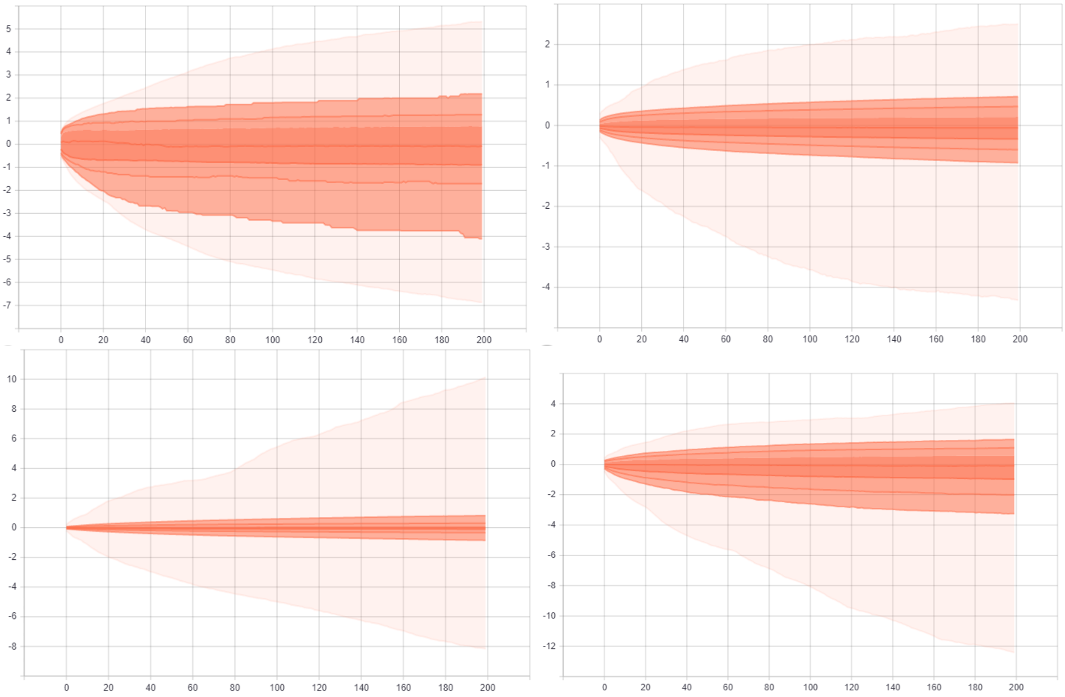
Wrist data set Distribution of kernel weights for the four layers of the network using Sigmoid at 200 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



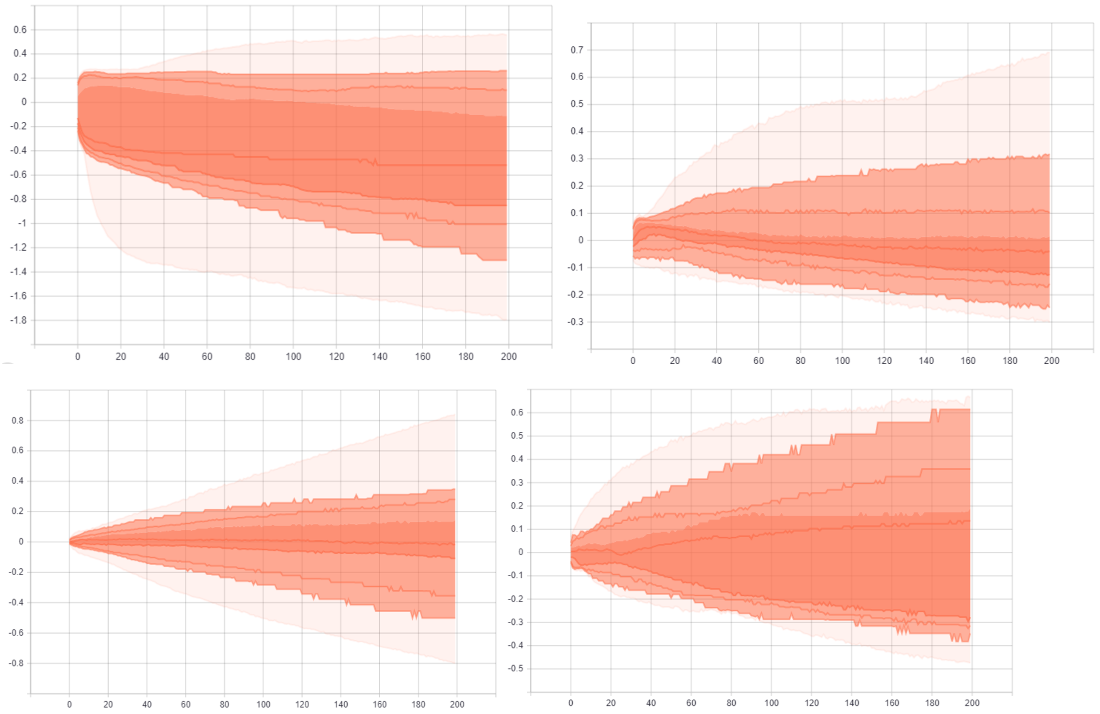
Wrist data set Distribution of bias weights for the four layers of the network using Sigmoid at 200 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



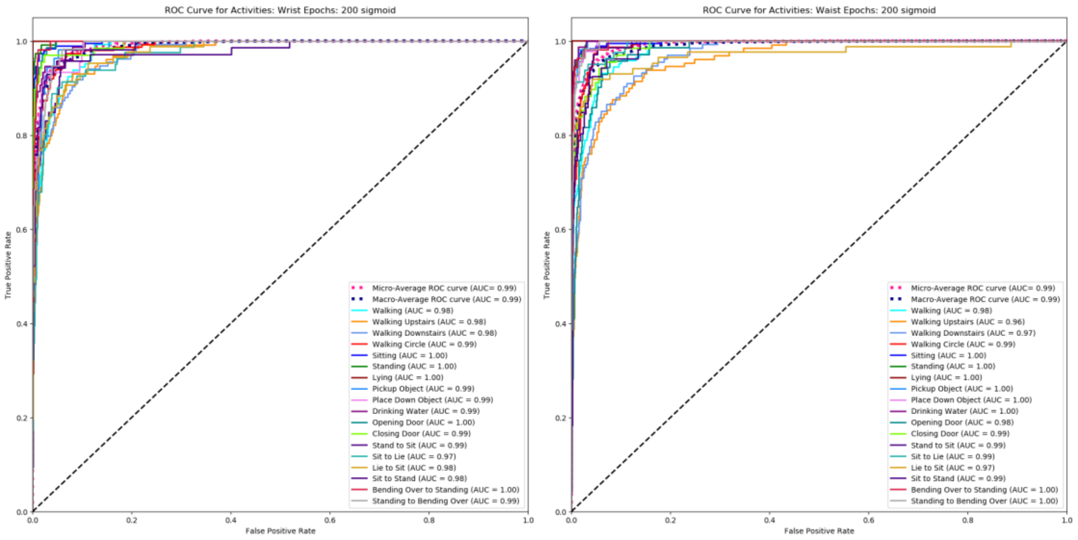
Waist data set Distribution of kernel weights for the four layers of the network using Sigmoid at 200 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



Waist data set Distribution of bias weights for the four layers of the network using Sigmoid at 200 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.

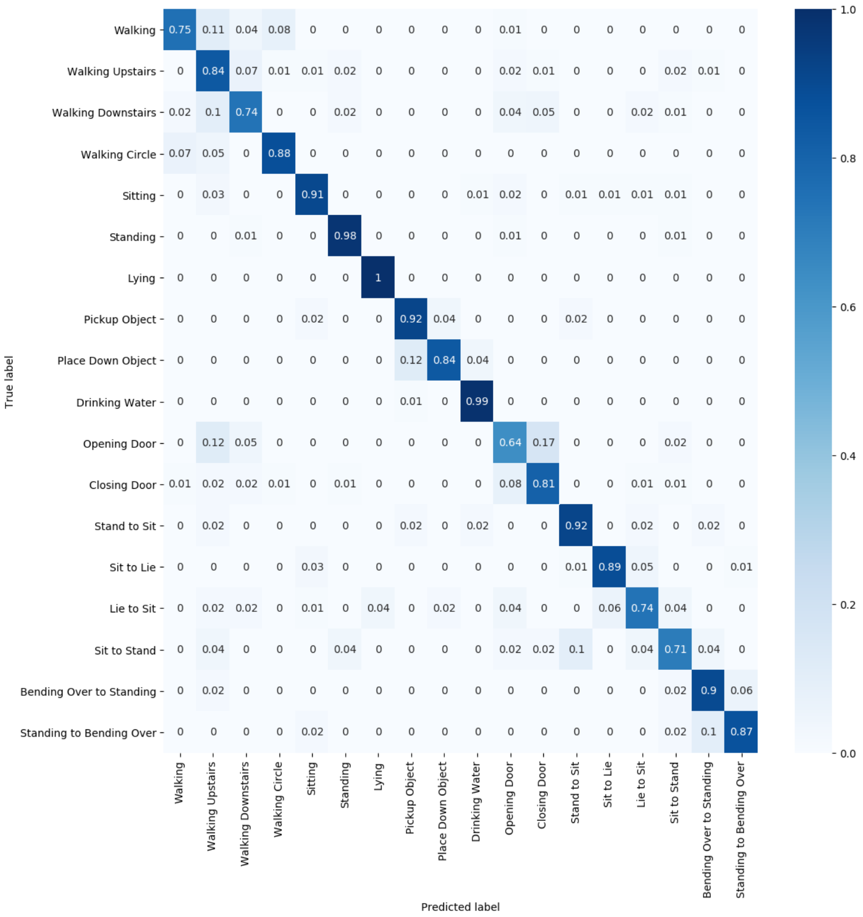


ROC Curve for wrist and waist

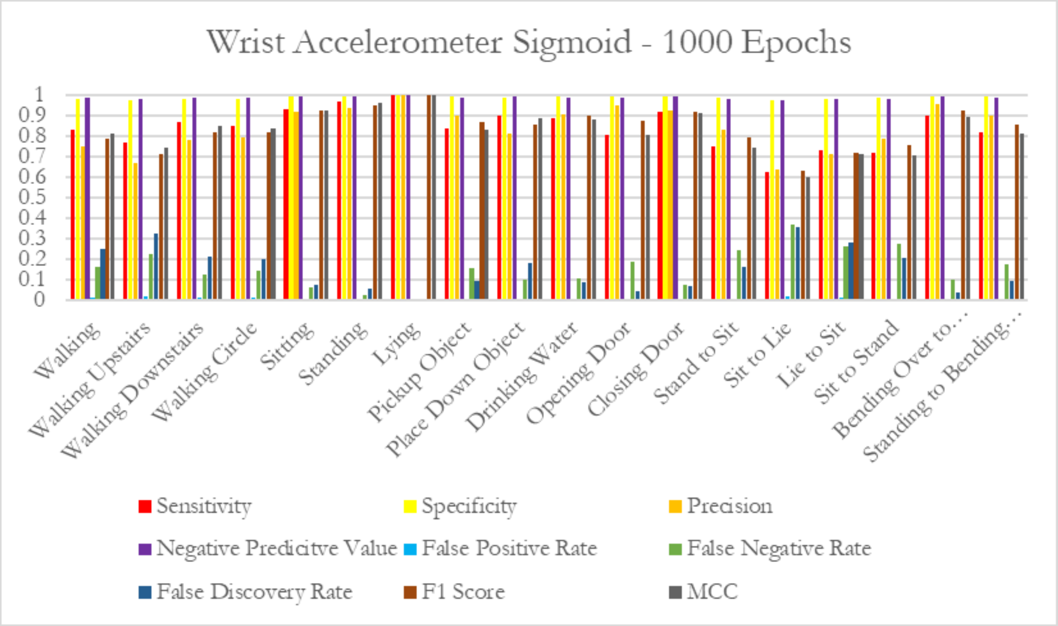


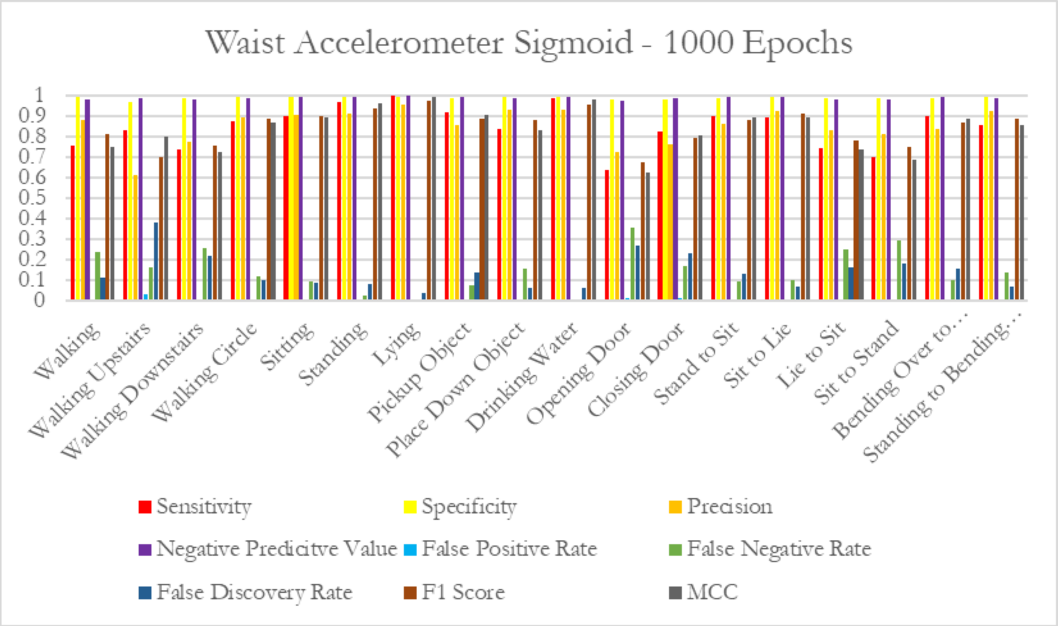
Appendix K – Additional Graphs: Sigmoid 1000 Epochs

Confusion matrix for waist

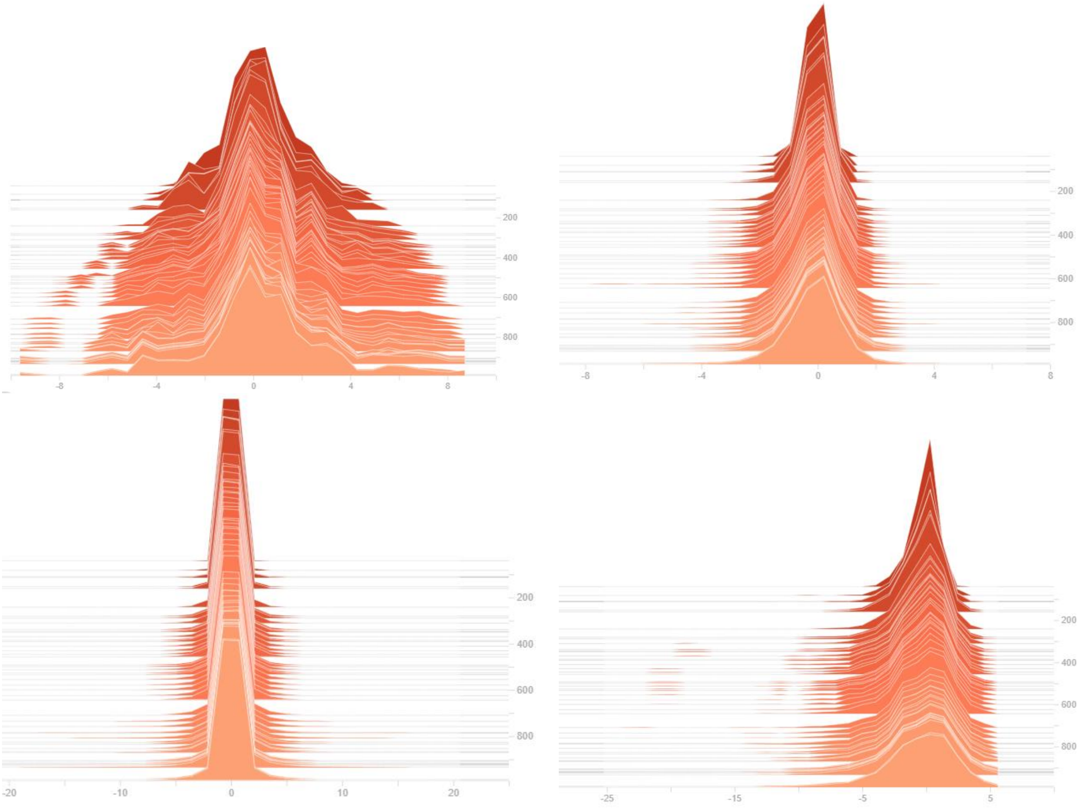


	Walking	Walking Upstairs	Walking Downstairs	Walking Circle	Sitting	Standing	Lying	Pickup Object	Place Down Object	Drinking Water	Opening Door	Closing Door	Stand to Sit	Sit to Lie	Lie to Sit	Sit to Stand	Bending Over to Standing	Standing to Bending Over
Walking	0.75	0.11	0.04	0.08	0	0	0	0	0	0	0.01	0	0	0	0	0	0	0
Walking Upstairs	0	0.84	0.07	0.01	0.01	0.02	0	0	0	0	0.02	0.01	0	0	0	0.02	0.01	0
Walking Downstairs	0.02	0.1	0.74	0	0	0.02	0	0	0	0	0.04	0.05	0	0	0.02	0.01	0	0
Walking Circle	0.07	0.05	0	0.88	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sitting	0	0.03	0	0	0.91	0	0	0	0	0.01	0.02	0	0.01	0.01	0.01	0.01	0	0
Standing	0	0	0.01	0	0	0.98	0	0	0	0	0.01	0	0	0	0	0.01	0	0
Lying	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Pickup Object	0	0	0	0	0.02	0	0	0.92	0.04	0	0	0	0.02	0	0	0	0	0
Place Down Object	0	0	0	0	0	0	0	0.12	0.84	0.04	0	0	0	0	0	0	0	0
Drinking Water	0	0	0	0	0	0	0	0.01	0	0.99	0	0	0	0	0	0	0	0
Opening Door	0	0.12	0.05	0	0	0	0	0	0	0	0.64	0.17	0	0	0	0.02	0	0
Closing Door	0.01	0.02	0.02	0.01	0	0.01	0	0	0	0	0.08	0.81	0	0	0.01	0.01	0	0
Stand to Sit	0	0.02	0	0	0	0	0	0.02	0	0.02	0	0	0.92	0	0.02	0	0.02	0
Sit to Lie	0	0	0	0	0.03	0	0	0	0	0	0	0	0.01	0.89	0.05	0	0	0.01
Lie to Sit	0	0.02	0.02	0	0.01	0	0.04	0	0.02	0	0.04	0	0	0.06	0.74	0.04	0	0
Sit to Stand	0	0.04	0	0	0	0.04	0	0	0	0	0.02	0.02	0.1	0	0.04	0.71	0.04	0
Bending Over to Standing	0	0.02	0	0	0	0	0	0	0	0	0	0	0	0	0	0.02	0.9	0.06
Standing to Bending Over	0	0	0	0	0.02	0	0	0	0	0	0	0	0	0	0	0.02	0.1	0.87

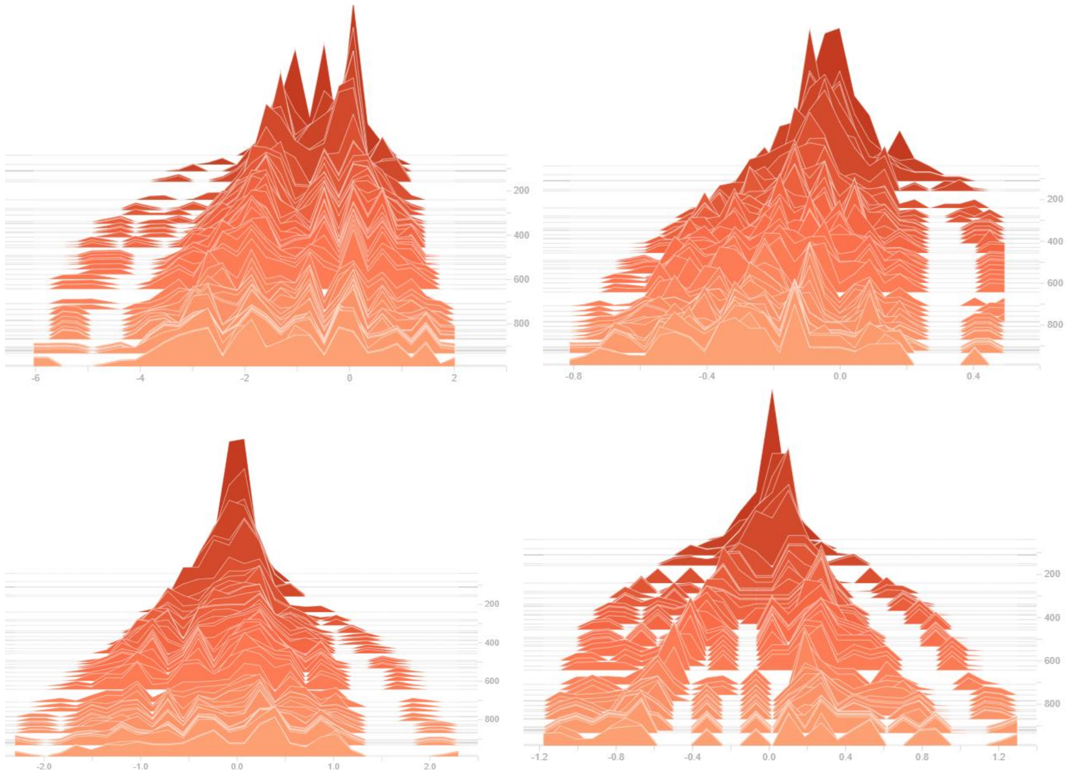




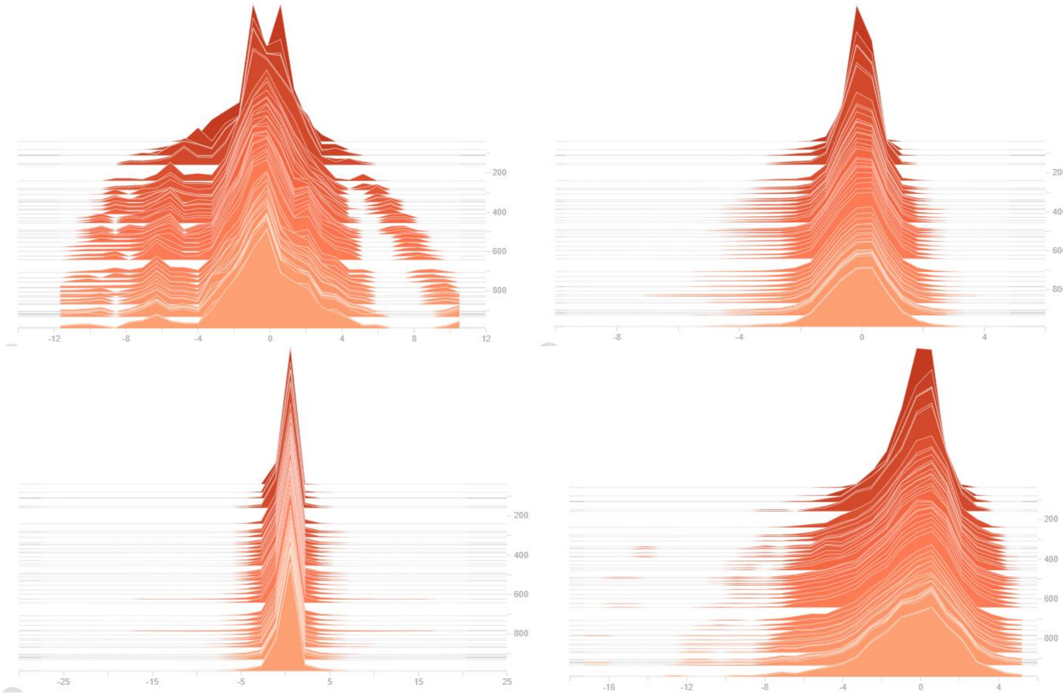
Wrist data set Histogram of kernel weights for the four layers of the network using Sigmoid at 1000 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



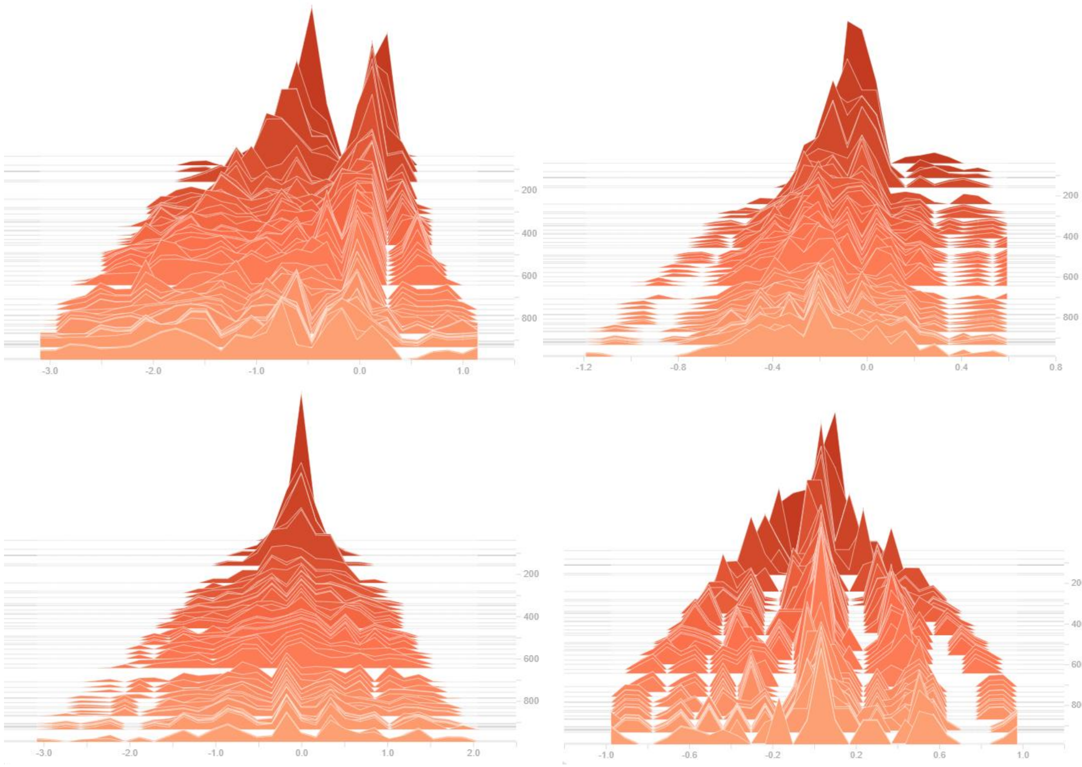
Wrist data set Histogram of bias weights for the four layers of the network using Sigmoid at 1000 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



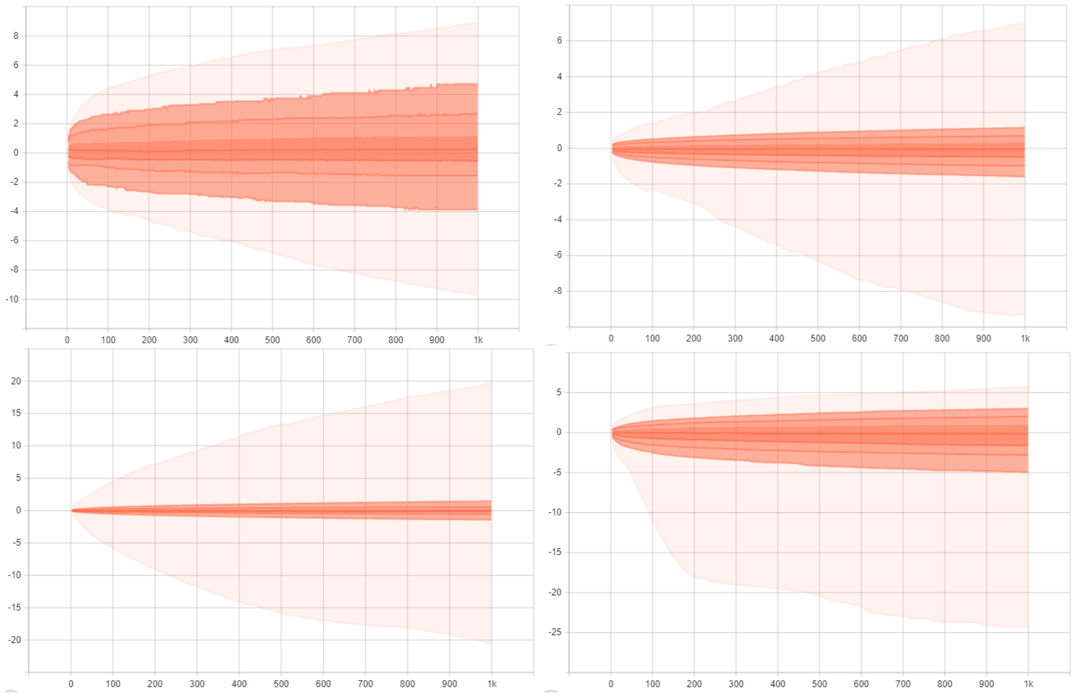
Waist data set Histogram of kernel weights for the four layers of the network using Sigmoid at 1000 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



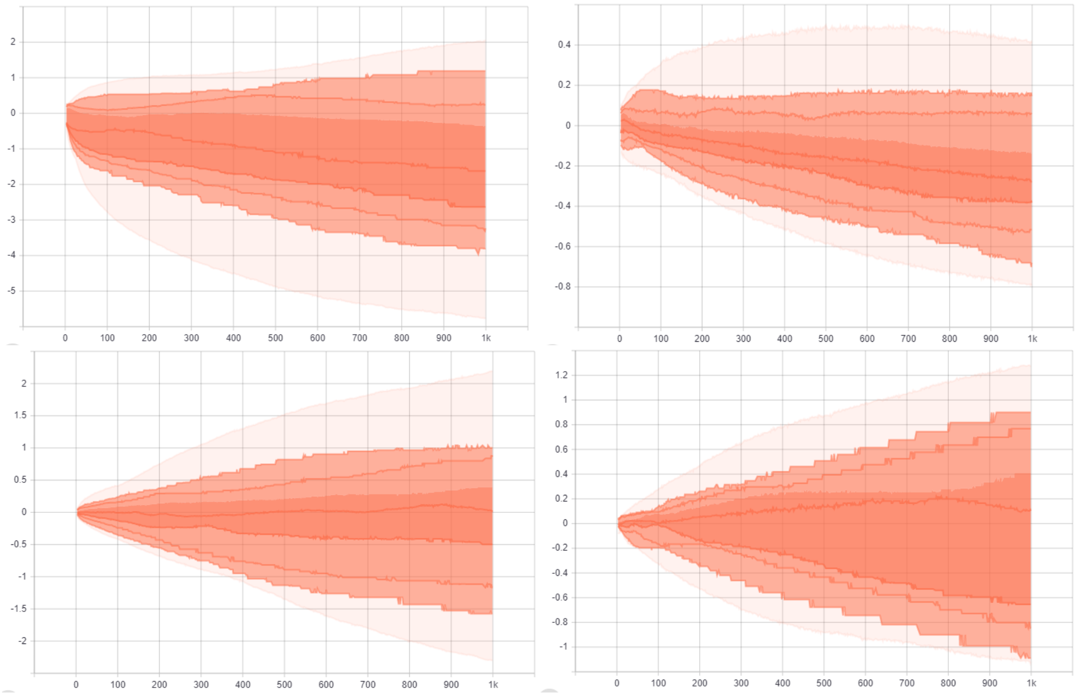
Waist data set Histogram of bias weights for the four layers of the network using Sigmoid at 1000 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



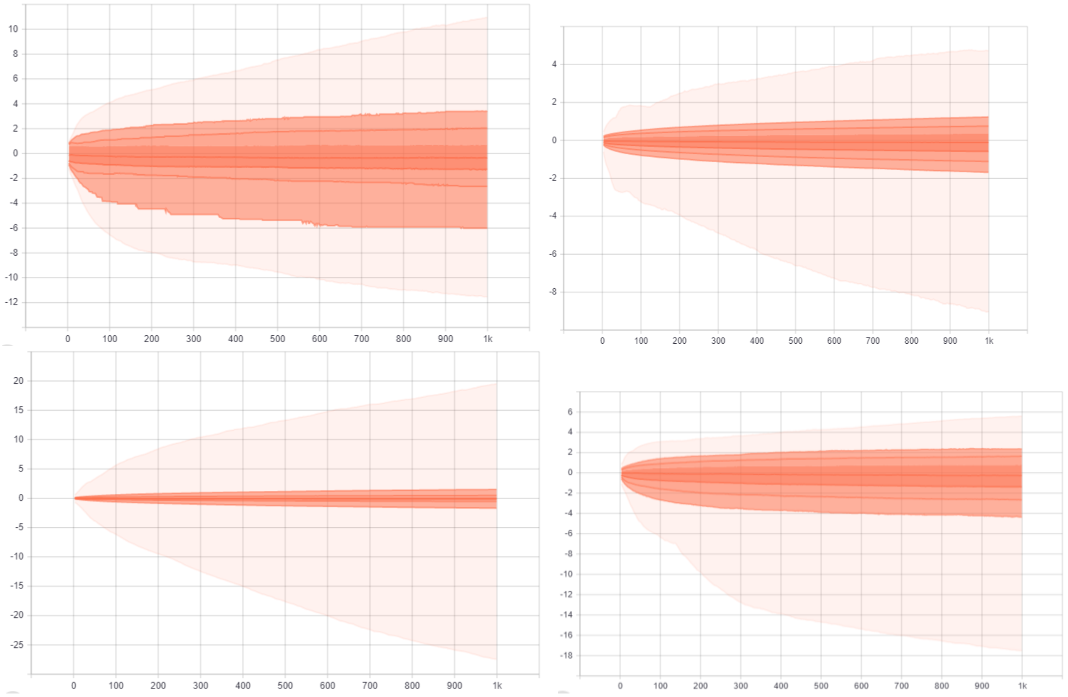
Wrist data set Distribution of kernel weights for the four layers of the network using Sigmoid at 1000 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



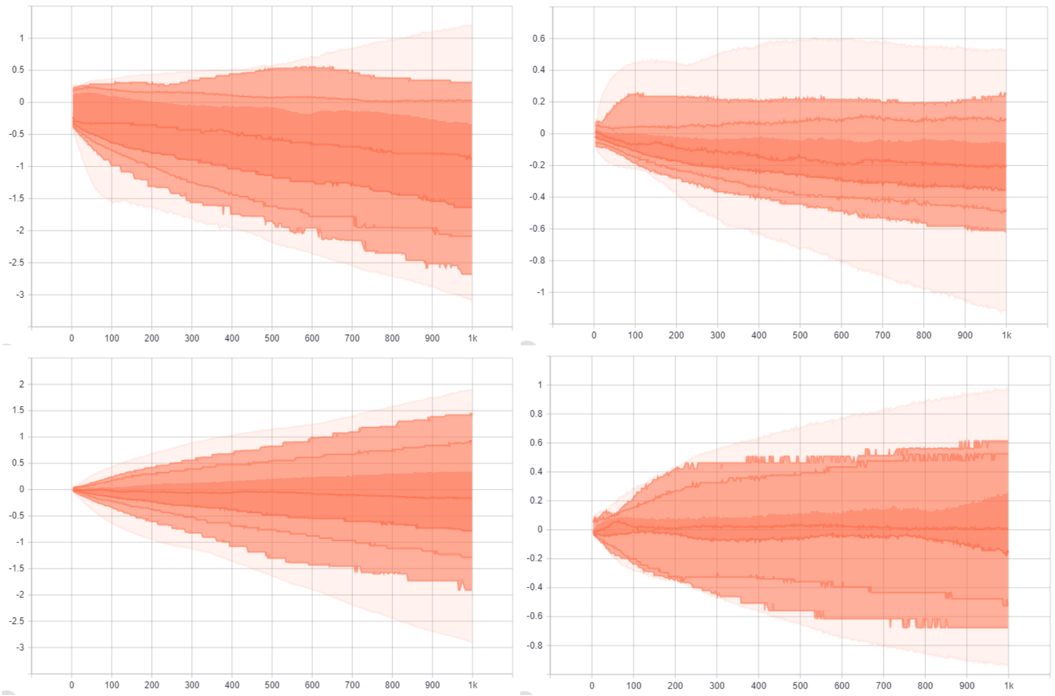
Wrist data set Distribution of bias weights for the four layers of the network using Sigmoid at 1000 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



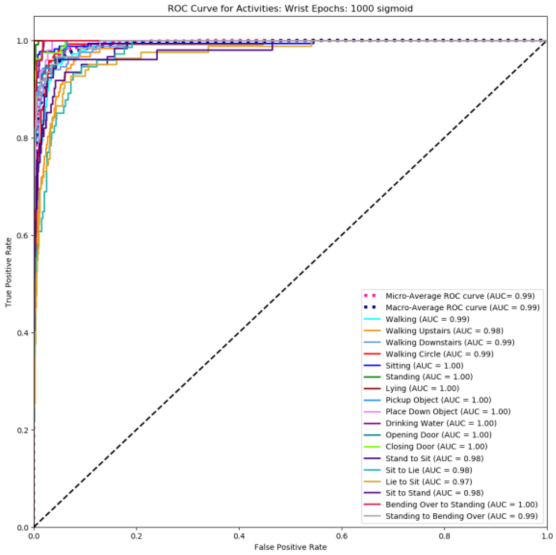
Waist data set Distribution of kernel weights for the four layers of the network using Sigmoid at 1000 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.

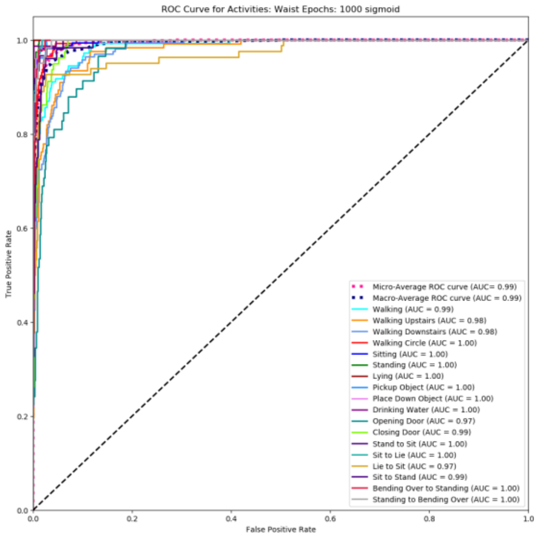


Waist data set Distribution of bias weights for the four layers of the network using Sigmoid at 1000 epochs. From top-1left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



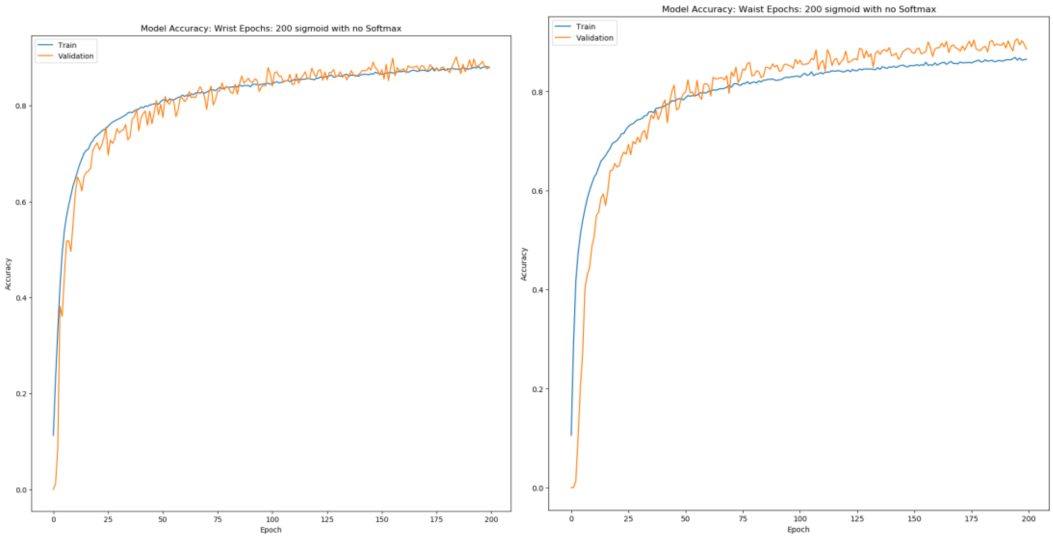
ROC Curve



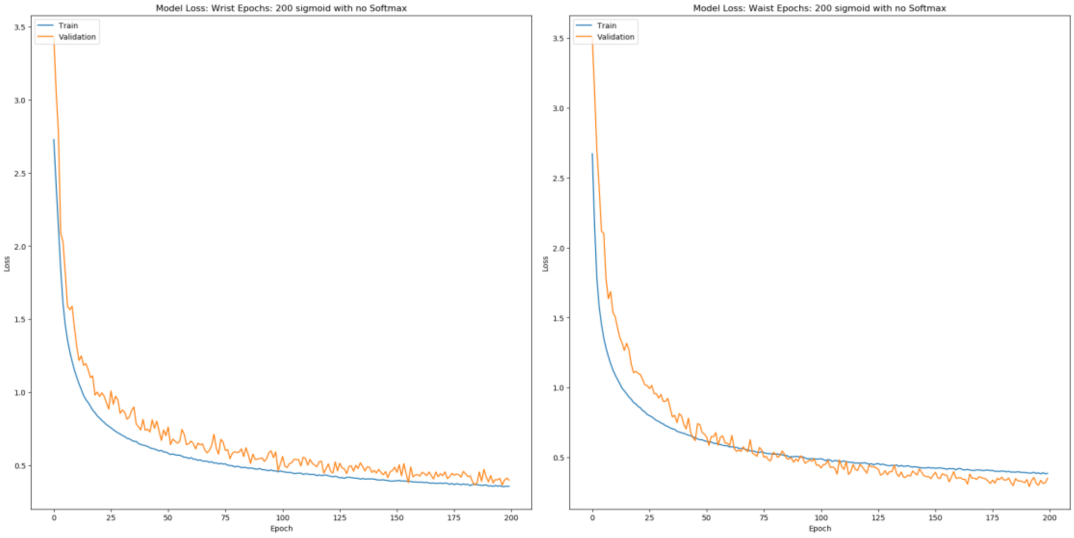


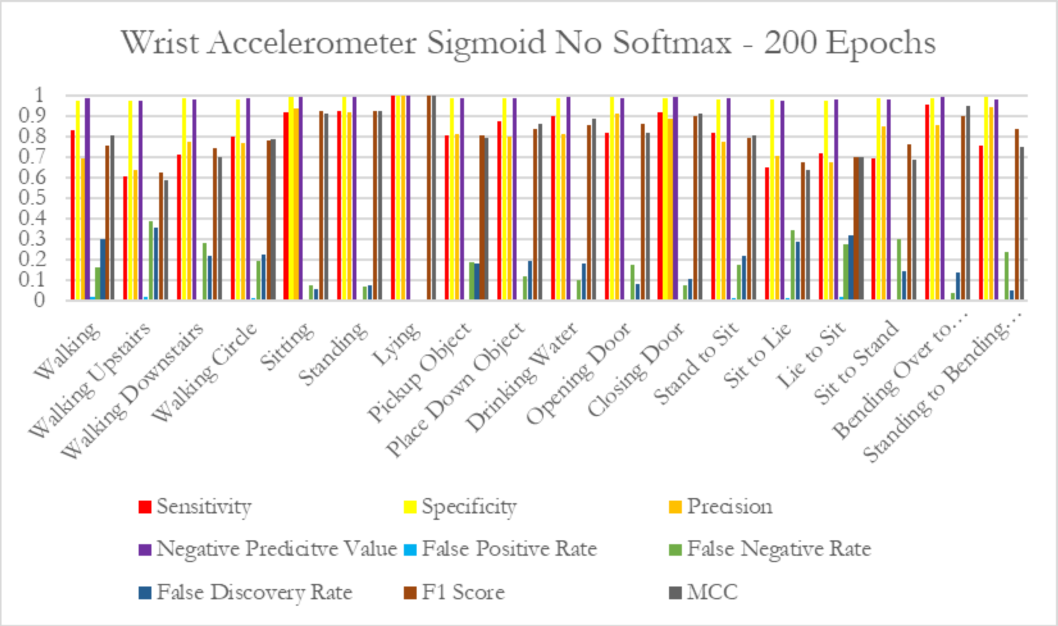
Appendix L – Additional Graphs: Sigmoid 200 Epochs No Softmax

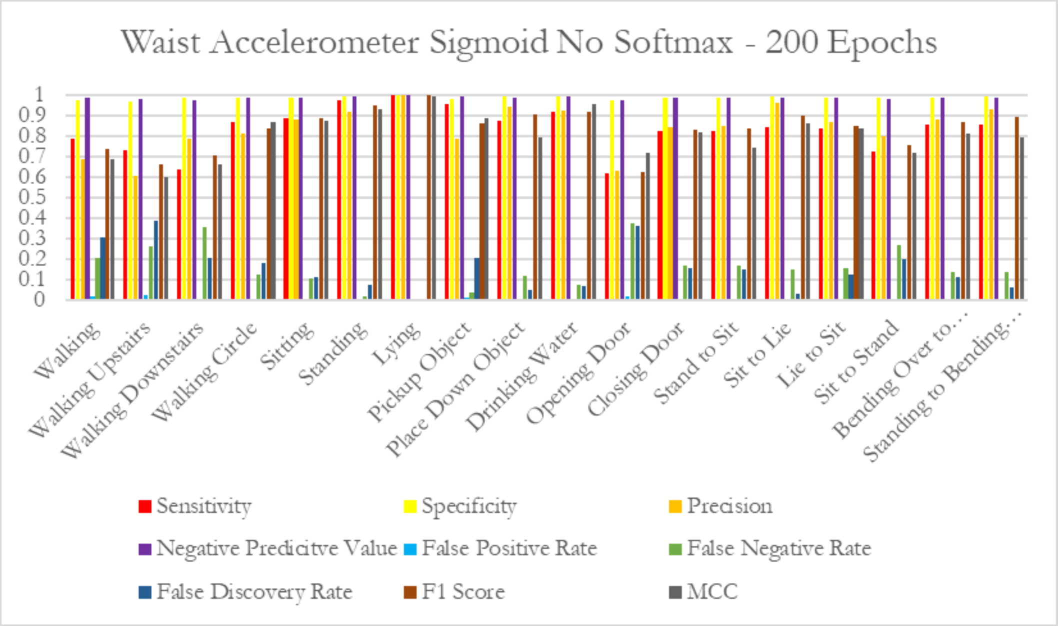
Wrist and Waist Accuracy



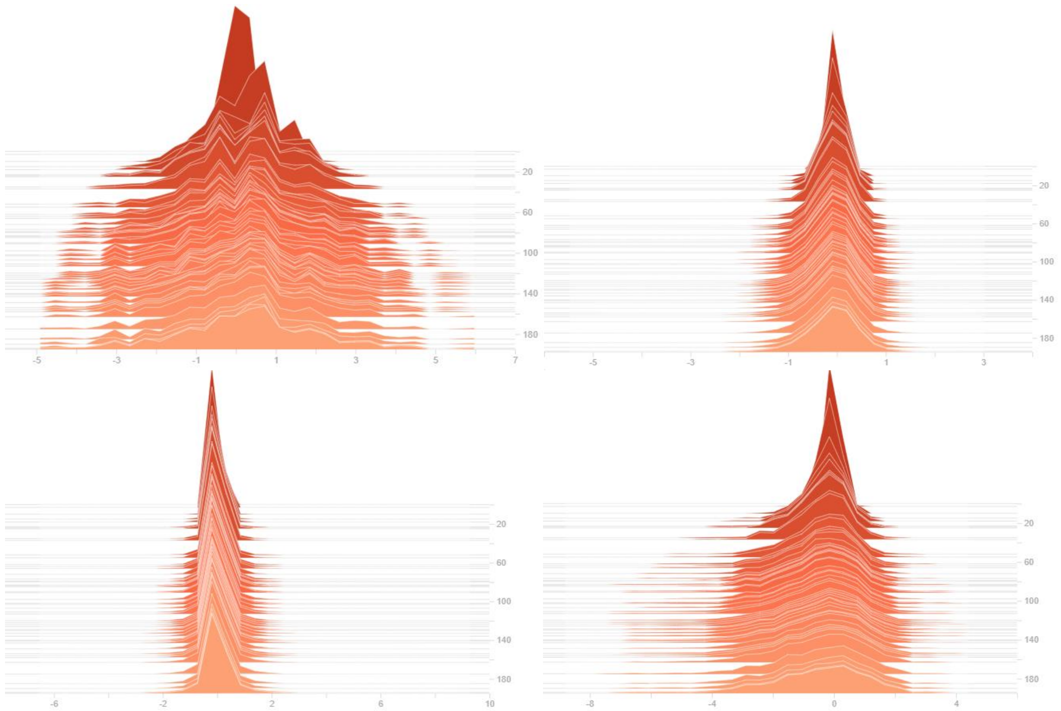
Wrist and Waist Loss



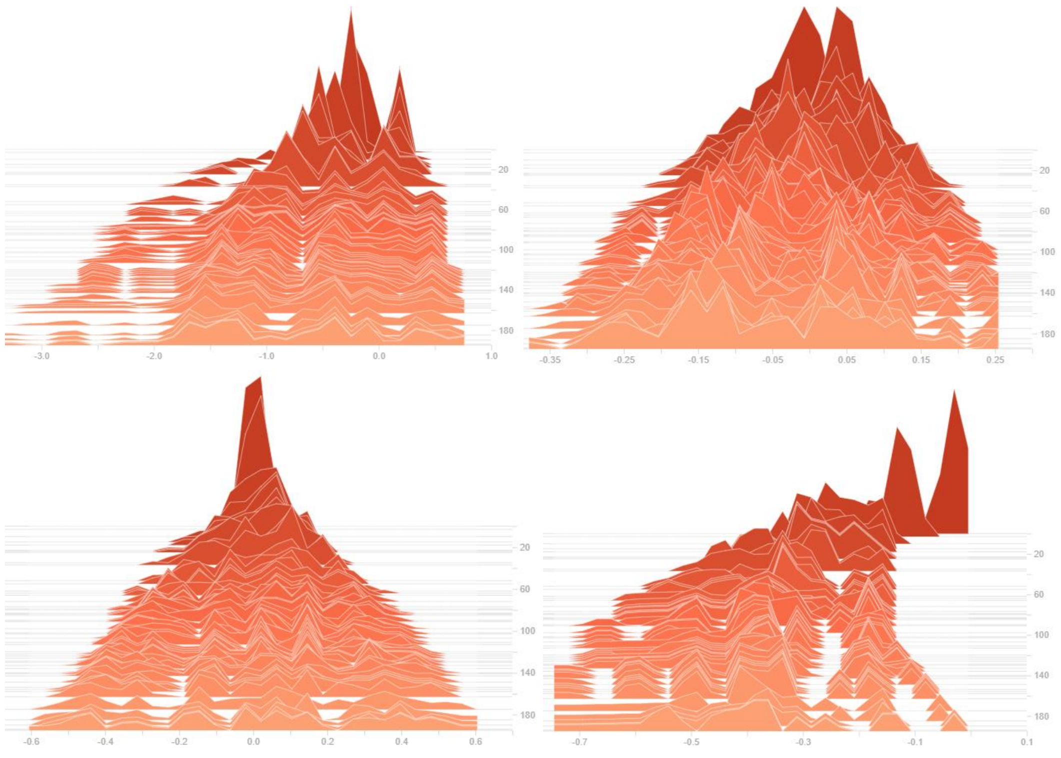

Wrist Accelerometer Sigmoid No Softmax - 200 Epochs
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Walking
Walking Upstairs
Walking Downstairs
Walking Circle
Sitting
Standing
Lying
Pickup Object
Place Down Object
Drinking Water
Opening Door
Closing Door
Stand to Sit
Sit to Lie
Lie to Sit
Sit to Stand
Bending Over to…
Standing to Bending…
Sensitivity
Specificity
Precision
Negative Predicitve Value
False Positive Rate
False Negative Rate
False Discovery Rate
F1 Score
MCC


Waist Accelerometer Sigmoid No Softmax - 200 Epochs
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Walking
Walking Upstairs
Walking Downstairs
Walking Circle
Sitting
Standing
Lying
Pickup Object
Place Down Object
Drinking Water
Opening Door
Closing Door
Stand to Sit
Sit to Lie
Lie to Sit
Sit to Stand
Bending Over to…
Standing to Bending…
Sensitivity
Specificity
Precision
Negative Predicitve Value
False Positive Rate
False Negative Rate
False Discovery Rate
F1 Score
MCC

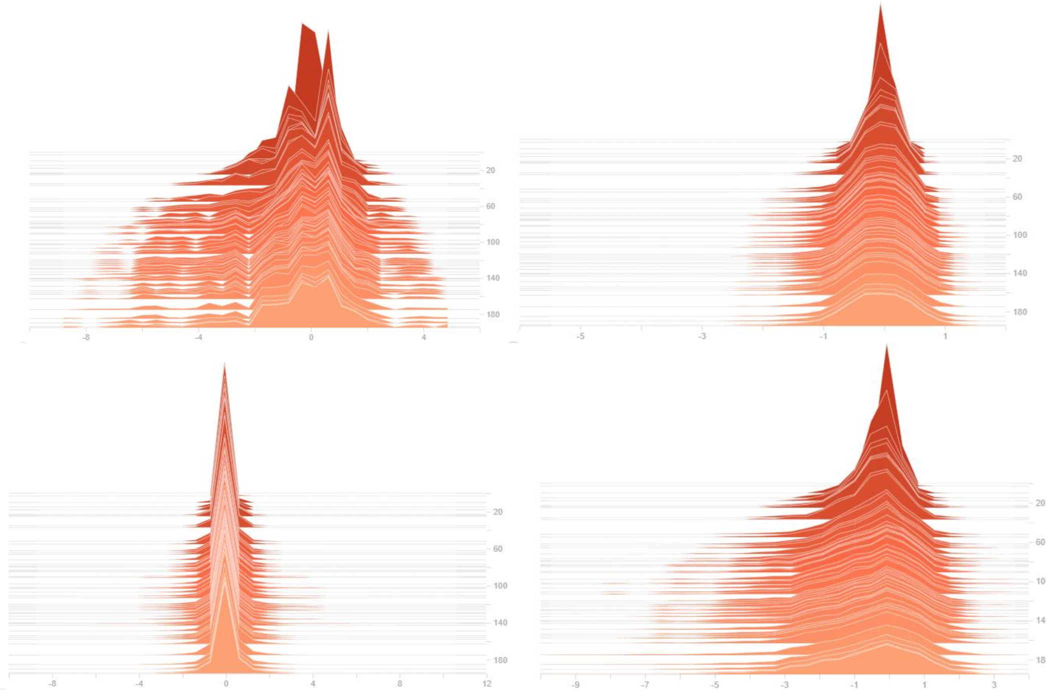
Wrist data set Histogram of kernel weights for the four layers of the network using Sigmoid at 200 epochs with no Softmax. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



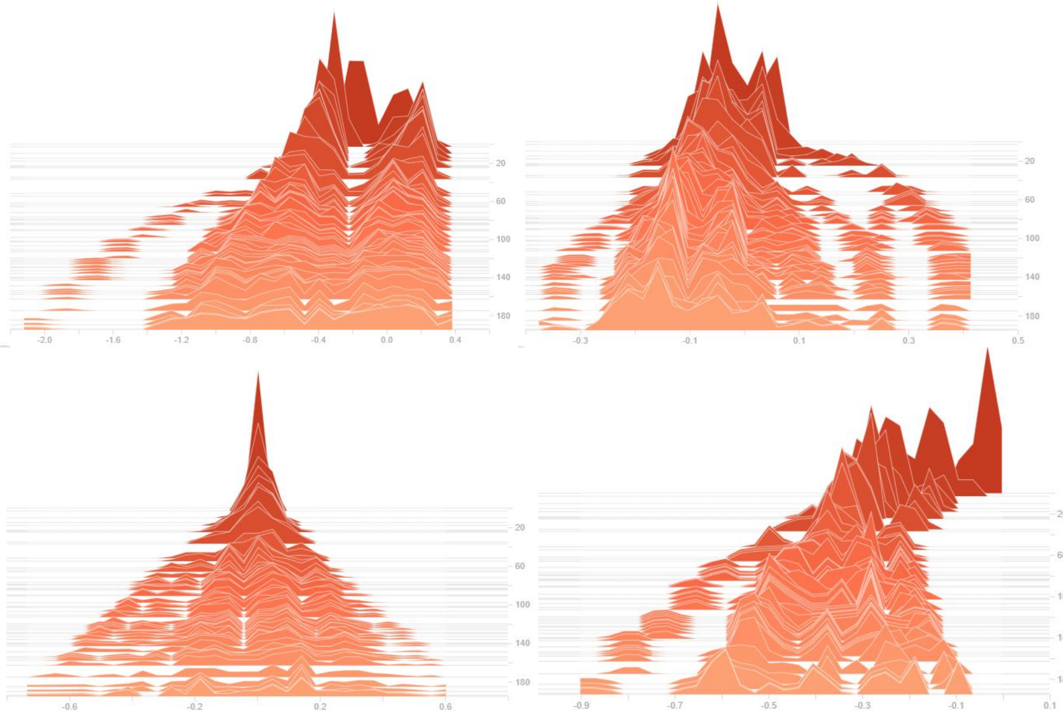
Wrist data set Histogram of bias weights for the four layers of the network using Sigmoid at 200 epochs with no Softmax. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



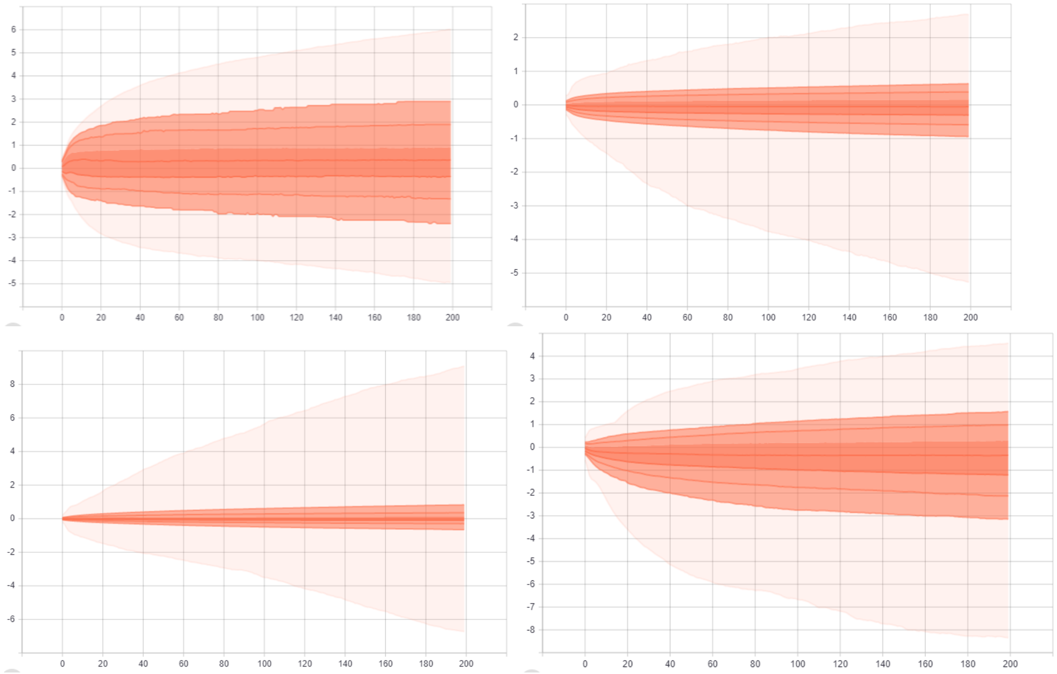
Waist data set Histogram of kernel weights for the four layers of the network using Sigmoid at 200 epochs with no Softmax. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



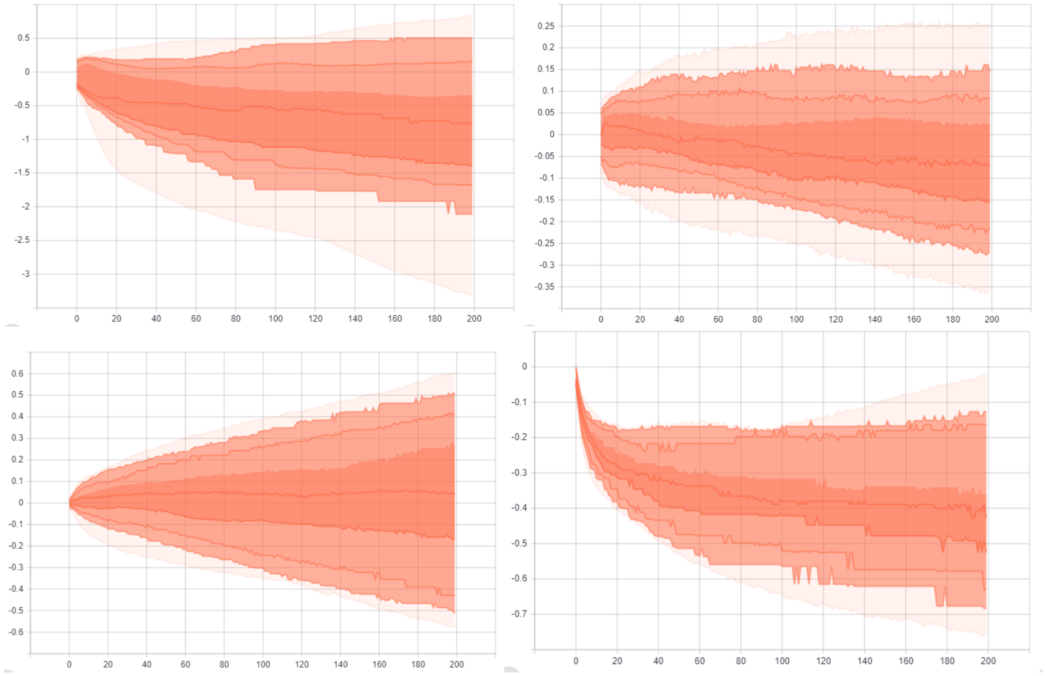
Waist data set Histogram of bias weights for the four layers of the network using Sigmoid at 200 epochs with no Softmax. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



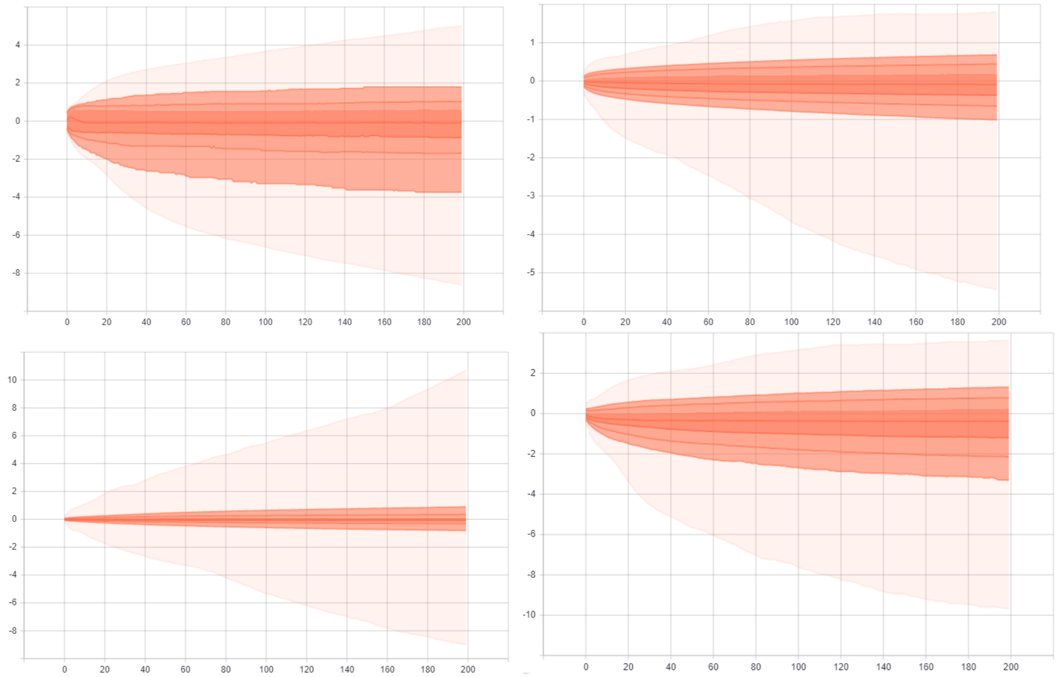
Wrist data set Distribution of kernel weights for the four layers of the network using Sigmoid at 200 epochs with no Softmax. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



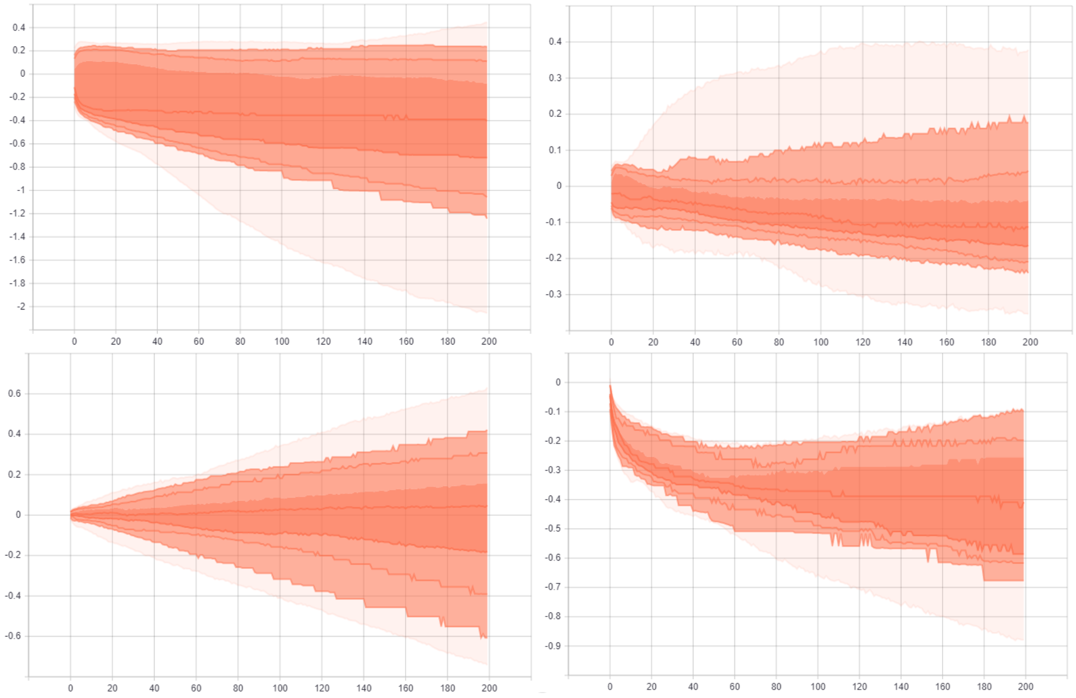
Wrist data set Distribution of bias weights for the four layers of the network using Sigmoid at 200 epochs with no Softmax. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



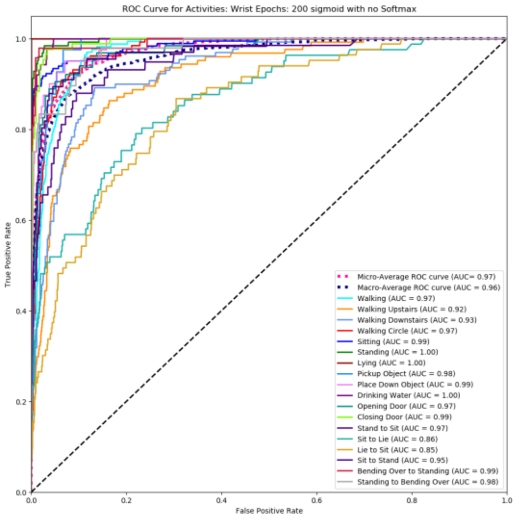
Waist data set Distribution of kernel weights for the four layers of the network using Sigmoid at 200 epochs with no Softmax. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.

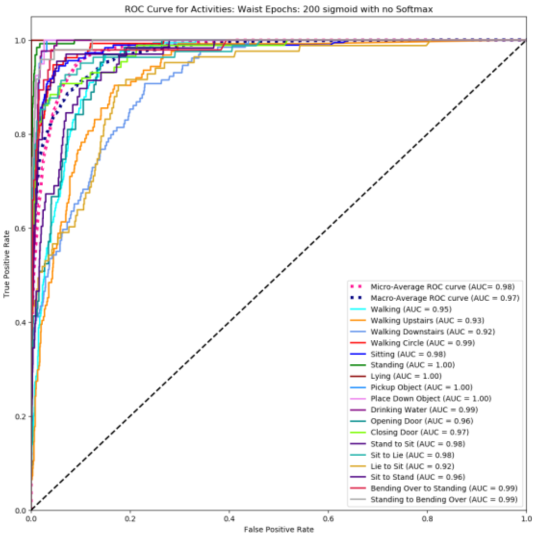


Waist data set Distribution of bias weights for the four layers of the network using Sigmoid at 200 epochs with no Softmax. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



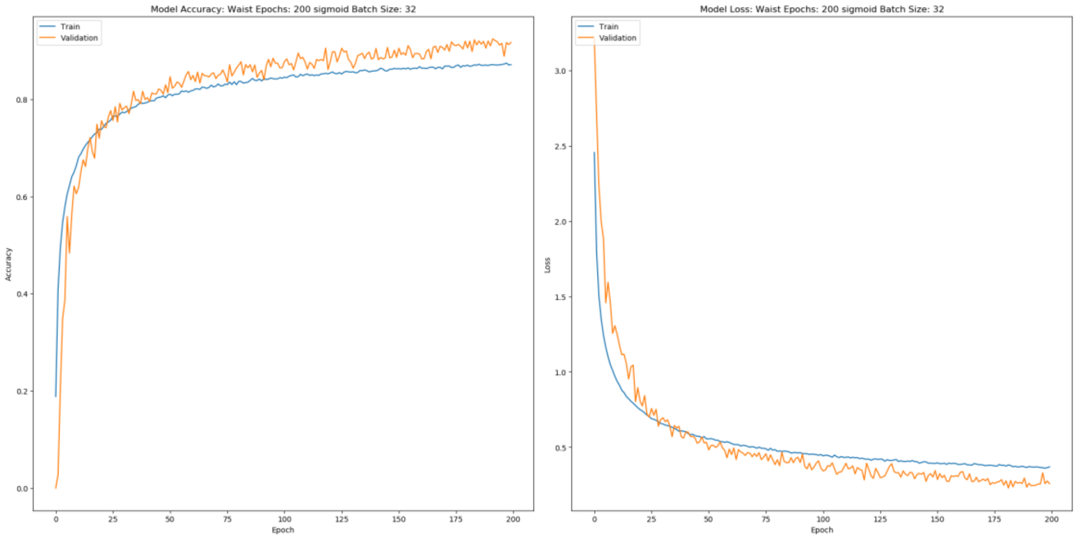
ROC Curve for Wrist and Waist



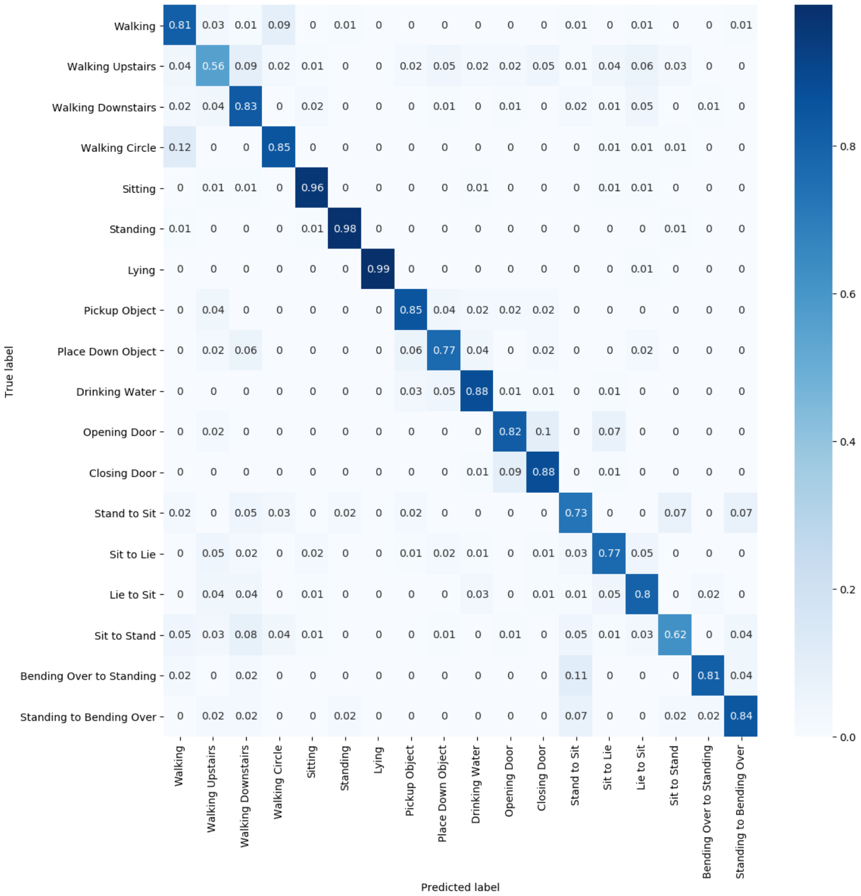


Appendix M – Additional Graphs: Sigmoid 200 Epochs Batch Size 32

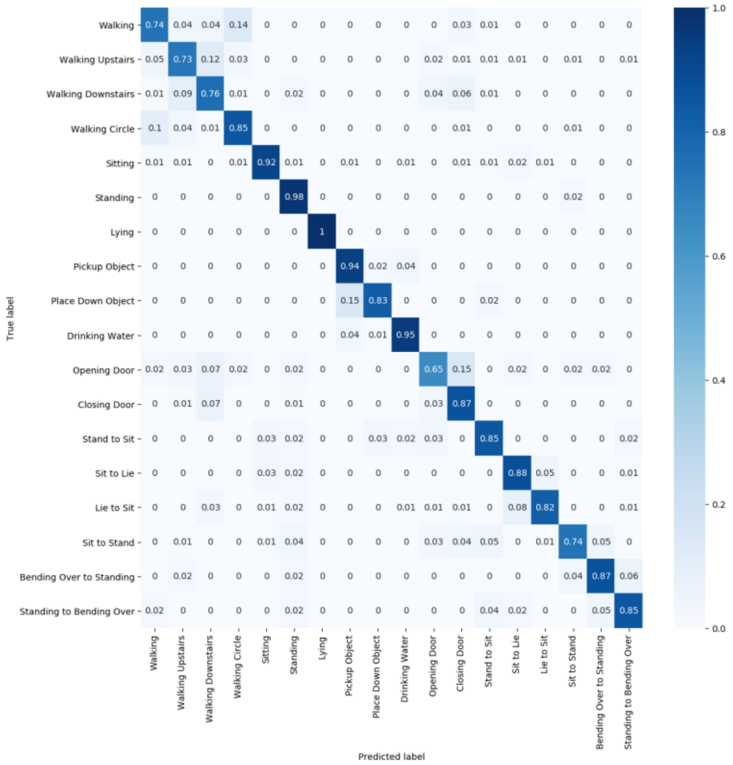
Waist accuracy and loss



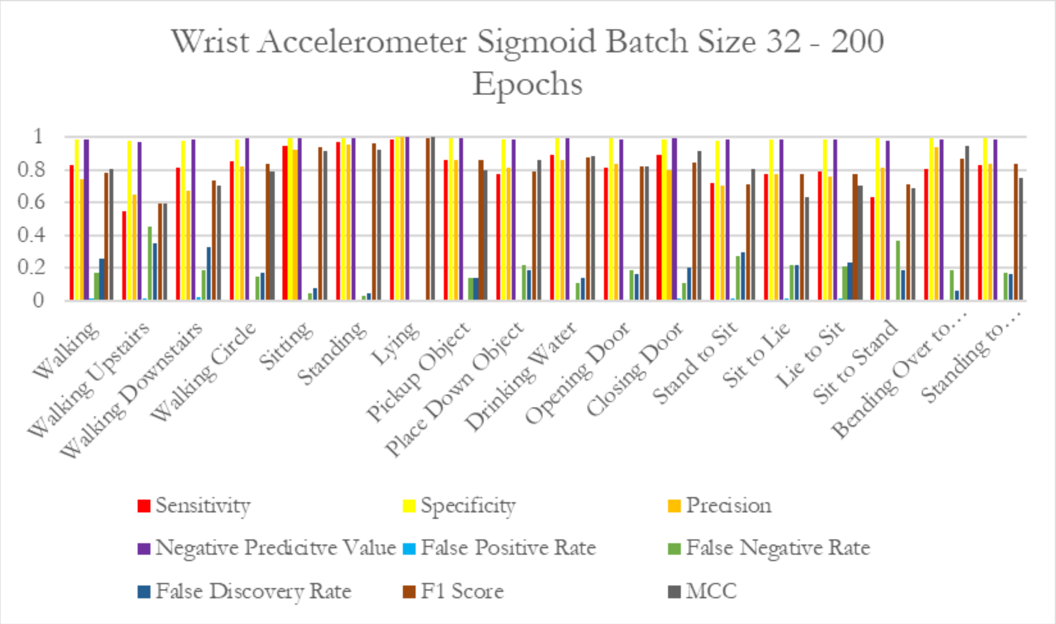
Wrist and waist confusion matrix

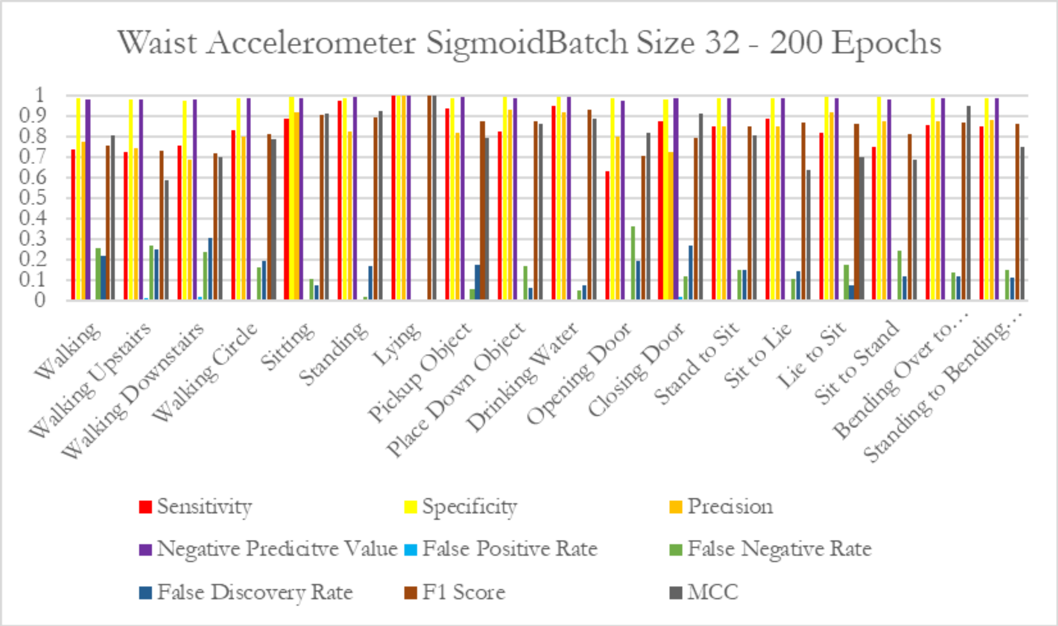


	Walking	Walking Upstairs	Walking Downstairs	Walking Circle	Sitting	Standing	Lying	Pickup Object	Place Down Object	Drinking Water	Opening Door	Closing Door	Stand to Sit	Sit to Lie	Lie to Sit	Sit to Stand	Bending Over to Standing	Standing to Bending Over
Walking	0.81	0.03	0.01	0.09	0	0.01	0	0	0	0	0	0	0.01	0	0.01	0	0	0.01
Walking Upstairs	0.04	0.56	0.09	0.02	0.01	0	0	0.02	0.05	0.02	0.02	0.05	0.01	0.04	0.06	0.03	0	0
Walking Downstairs	0.02	0.04	0.83	0	0.02	0	0	0	0.01	0	0.01	0	0.02	0.01	0.05	0	0.01	0
Walking Circle	0.12	0	0	0.85	0	0	0	0	0	0	0	0	0	0.01	0.01	0.01	0	0
Sitting	0	0.01	0.01	0	0.96	0	0	0	0	0.01	0	0	0	0.01	0.01	0	0	0
Standing	0.01	0	0	0	0.01	0.98	0	0	0	0	0	0	0	0	0	0.01	0	0
Lying	0	0	0	0	0	0	0.99	0	0	0	0	0	0	0	0.01	0	0	0
Pickup Object	0	0.04	0	0	0	0	0	0.85	0.04	0.02	0.02	0.02	0	0	0	0	0	0
Place Down Object	0	0.02	0.06	0	0	0	0	0.06	0.77	0.04	0	0.02	0	0	0.02	0	0	0
Drinking Water	0	0	0	0	0	0	0	0.03	0.05	0.88	0.01	0.01	0	0.01	0	0	0	0
Opening Door	0	0.02	0	0	0	0	0	0	0	0	0.82	0.1	0	0.07	0	0	0	0
Closing Door	0	0	0	0	0	0	0	0	0	0.01	0.09	0.88	0	0.01	0	0	0	0
Stand to Sit	0.02	0	0.05	0.03	0	0.02	0	0.02	0	0	0	0	0.73	0	0	0.07	0	0.07
Sit to Lie	0	0.05	0.02	0	0.02	0	0	0.01	0.02	0.01	0	0.01	0.03	0.77	0.05	0	0	0
Lie to Sit	0	0.04	0.04	0	0.01	0	0	0	0	0.03	0	0.01	0.01	0.05	0.8	0	0.02	0
Sit to Stand	0.05	0.03	0.08	0.04	0.01	0	0	0	0.01	0	0.01	0	0.05	0.01	0.03	0.62	0	0.04
Bending Over to Standing	0.02	0	0.02	0	0	0	0	0	0	0	0	0	0.11	0	0	0	0.81	0.04
Standing to Bending Over	0	0.02	0.02	0	0	0.02	0	0	0	0	0	0	0.07	0	0	0.02	0.02	0.84

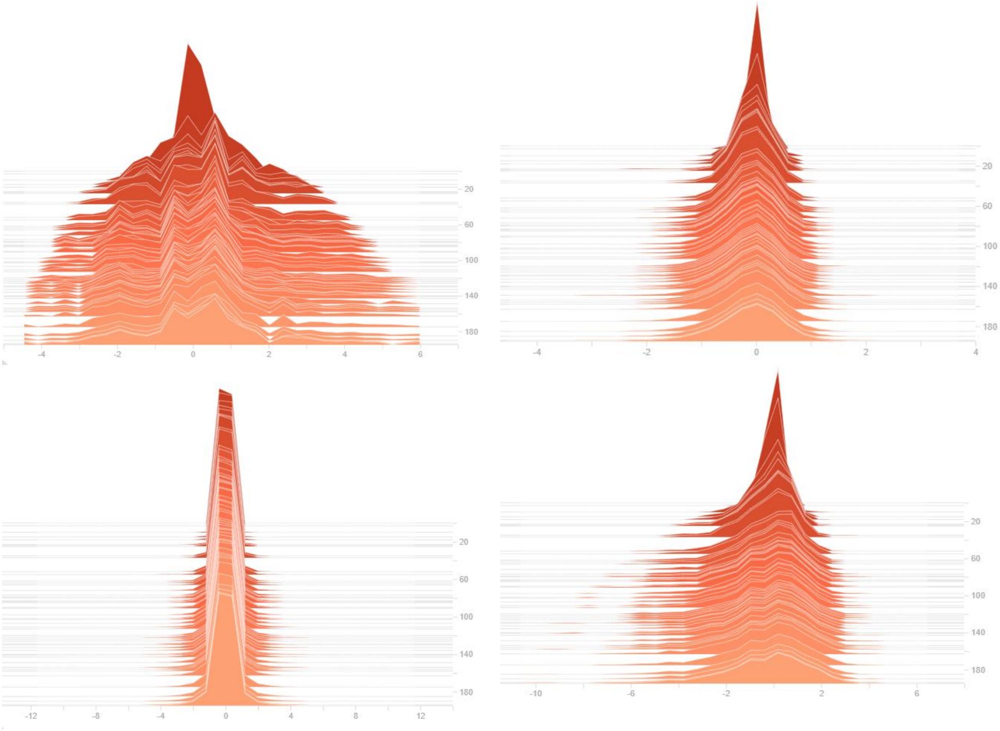


	Walking	Walking Upstairs	Walking Downstairs	Walking Circle	Sitting	Standing	Lying	Pickup Object	Place Down Object	Drinking Water	Opening Door	Closing Door	Stand to Sit	Sit to Lie	Lie to Sit	Sit to Stand	Bending Over to Standing	Standing to Bending Over
Walking	0.74	0.04	0.04	0.14	0	0	0	0	0	0	0	0.03	0.01	0	0	0	0	0
Walking Upstairs	0.05	0.73	0.12	0.03	0	0	0	0	0	0	0.02	0.01	0.01	0.01	0	0.01	0	0.01
Walking Downstairs	0.01	0.09	0.76	0.01	0	0.02	0	0	0	0	0.04	0.06	0.01	0	0	0	0	0
Walking Circle	0.1	0.04	0.01	0.85	0	0	0	0	0	0	0	0.01	0	0	0	0.01	0	0
Sitting	0.01	0.01	0	0.01	0.92	0.01	0	0.01	0	0.01	0	0.01	0.01	0.02	0.01	0	0	0
Standing	0	0	0	0	0	0.98	0	0	0	0	0	0	0	0	0	0.02	0	0
Lying	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Pickup Object	0	0	0	0	0	0	0	0.94	0.02	0.04	0	0	0	0	0	0	0	0
Place Down Object	0	0	0	0	0	0	0	0.15	0.83	0	0	0	0.02	0	0	0	0	0
Drinking Water	0	0	0	0	0	0	0	0.04	0.01	0.95	0	0	0	0	0	0	0	0
Opening Door	0.02	0.03	0.07	0.02	0	0.02	0	0	0	0	0.65	0.15	0	0.02	0	0.02	0.02	0
Closing Door	0	0.01	0.07	0	0	0.01	0	0	0	0	0.03	0.87	0	0	0	0	0	0
Stand to Sit	0	0	0	0	0.03	0.02	0	0	0.03	0.02	0.03	0	0.85	0	0	0	0	0.02
Sit to Lie	0	0	0	0	0.03	0.02	0	0	0	0	0	0	0	0.88	0.05	0	0	0.01
Lie to Sit	0	0	0.03	0	0.01	0.02	0	0	0	0.01	0.01	0.01	0	0.08	0.82	0	0	0.01
Sit to Stand	0	0.01	0	0	0.01	0.04	0	0	0	0	0.03	0.04	0.05	0	0.01	0.74	0.05	0
Bending Over to Standing	0	0.02	0	0	0	0.02	0	0	0	0	0	0	0	0	0	0.04	0.87	0.06
Standing to Bending Over	0.02	0	0	0	0	0.02	0	0	0	0	0	0	0.04	0.02	0	0	0.05	0.85

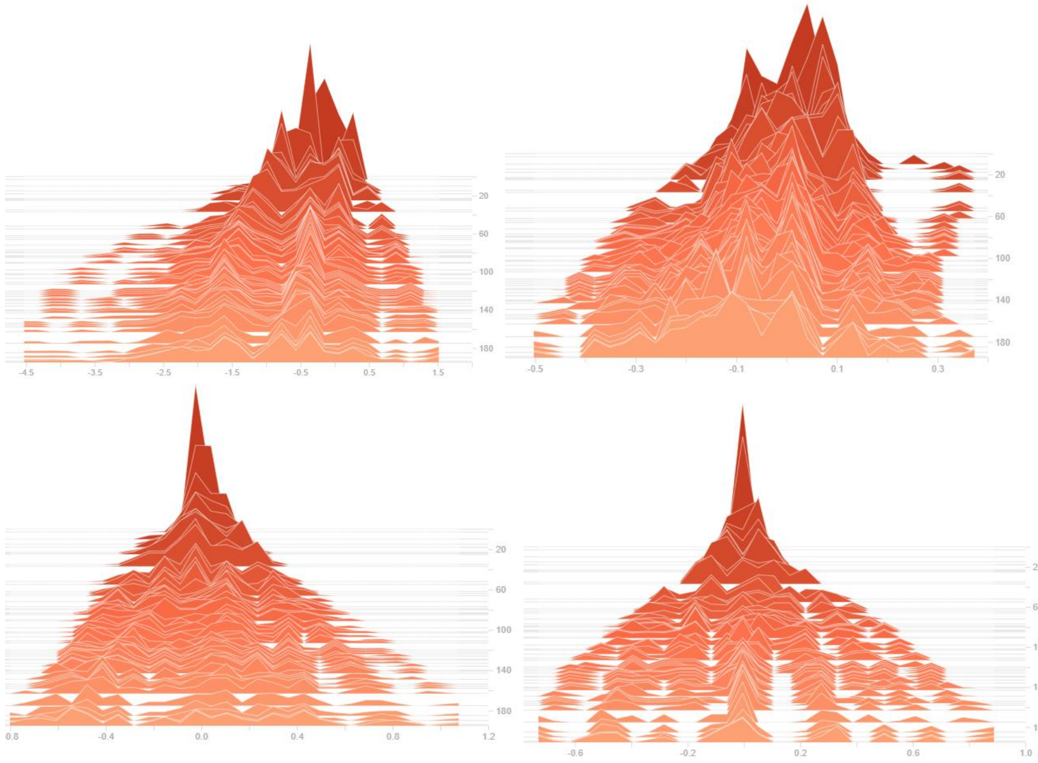




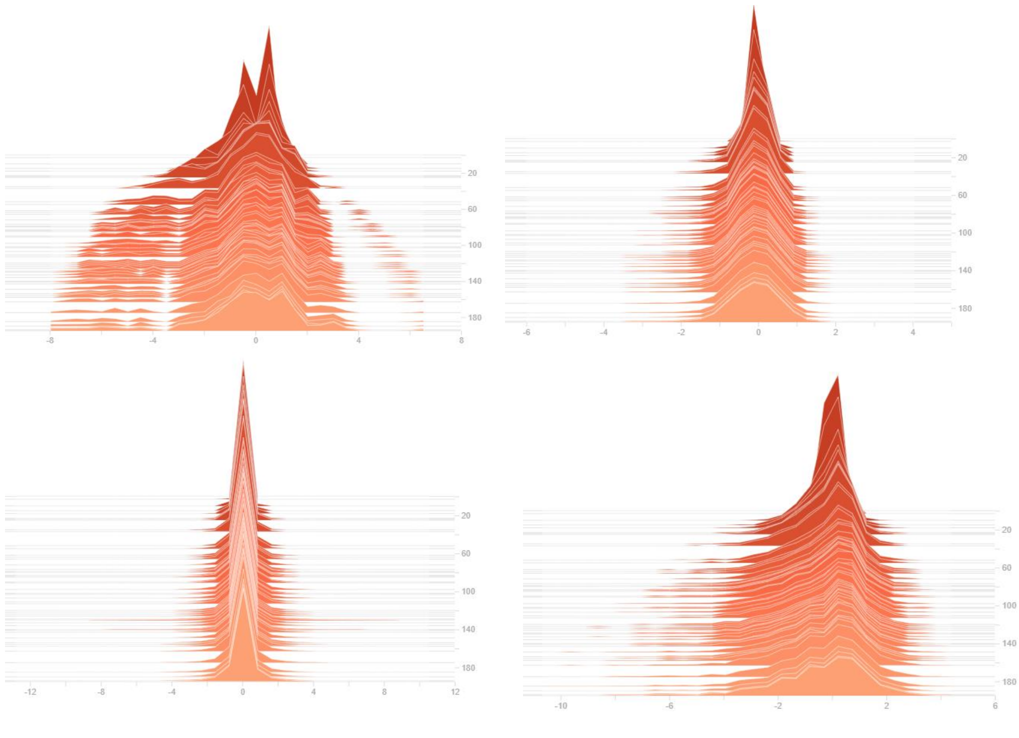
Wrist data set Histogram of kernel weights for the four layers of the network using Sigmoid at 200 epochs with batch size 32. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



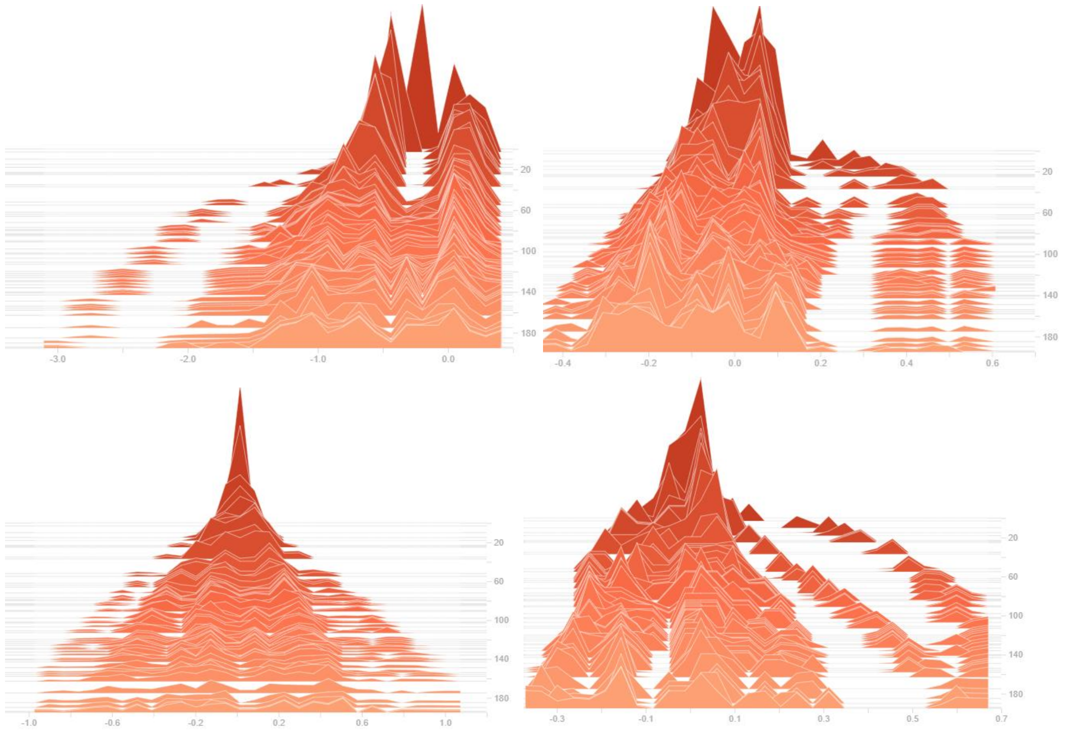
Wrist data set Histogram of bias weights for the four layers of the network using Sigmoid at 200 epochs with batch size 32. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



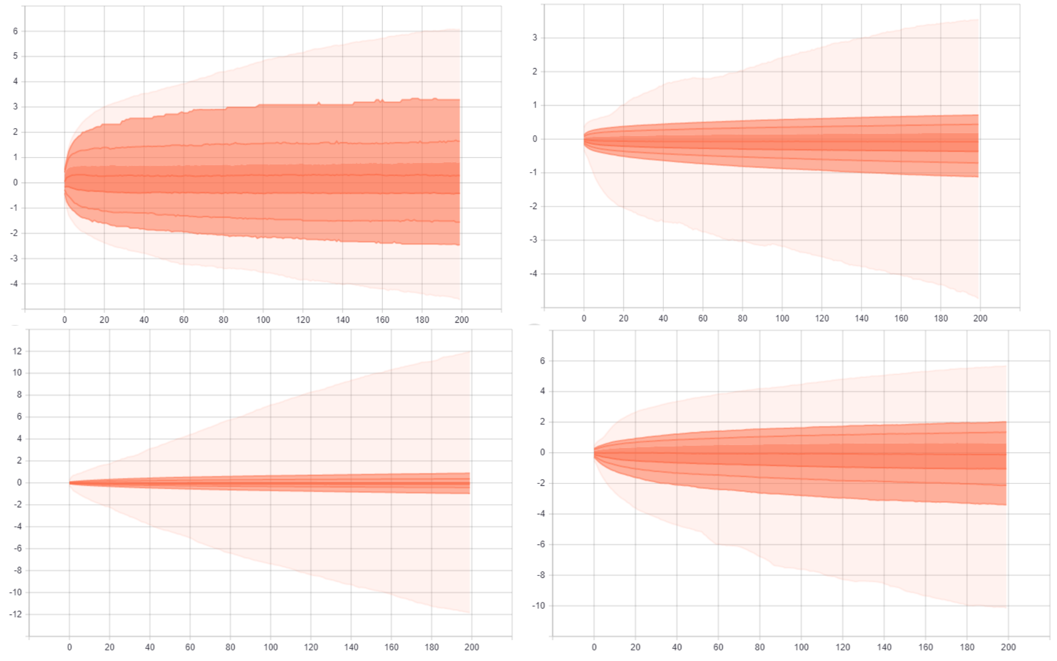
Waist data set Histogram of kernel weights for the four layers of the network using Sigmoid at 200 epochs with batch size 32. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



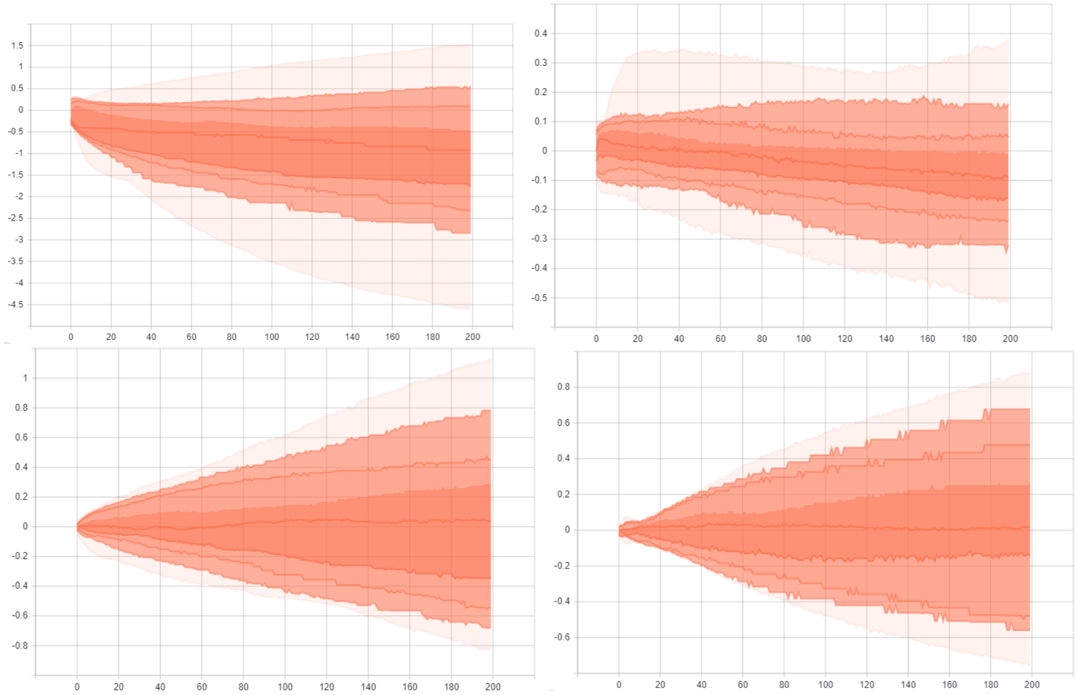
Waist data set Histogram of bias weights for the four layers of the network using Sigmoid at 200 epochs with batch size 32. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



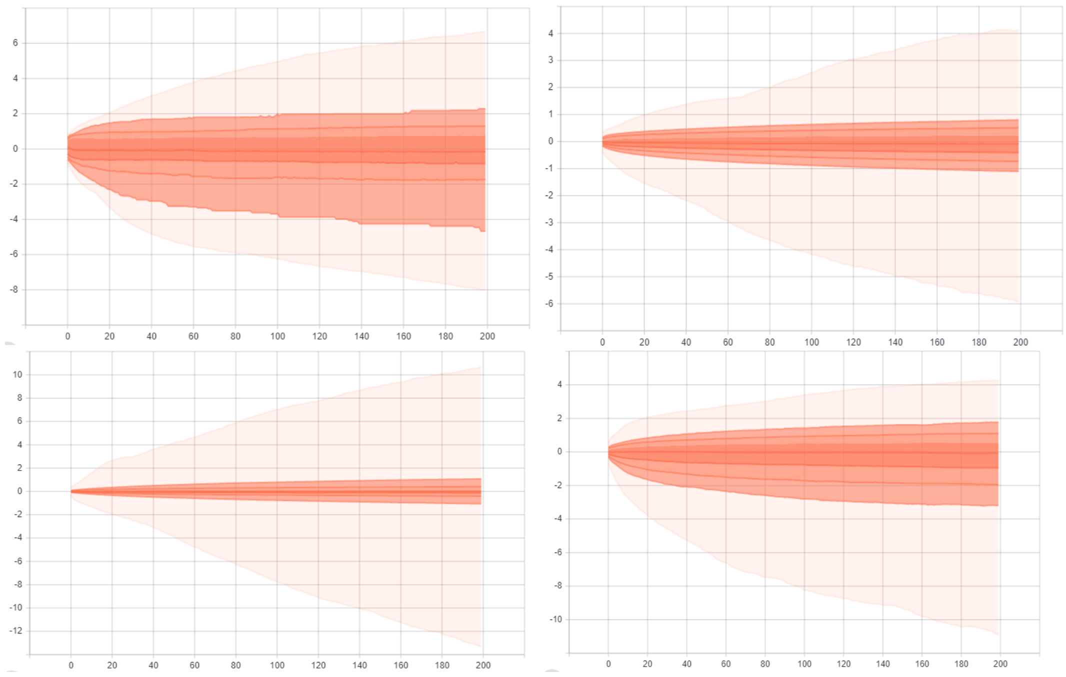
Wrist data set Distribution of kernel weights for the four layers of the network using Sigmoid at 200 epochs with batch size 32. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



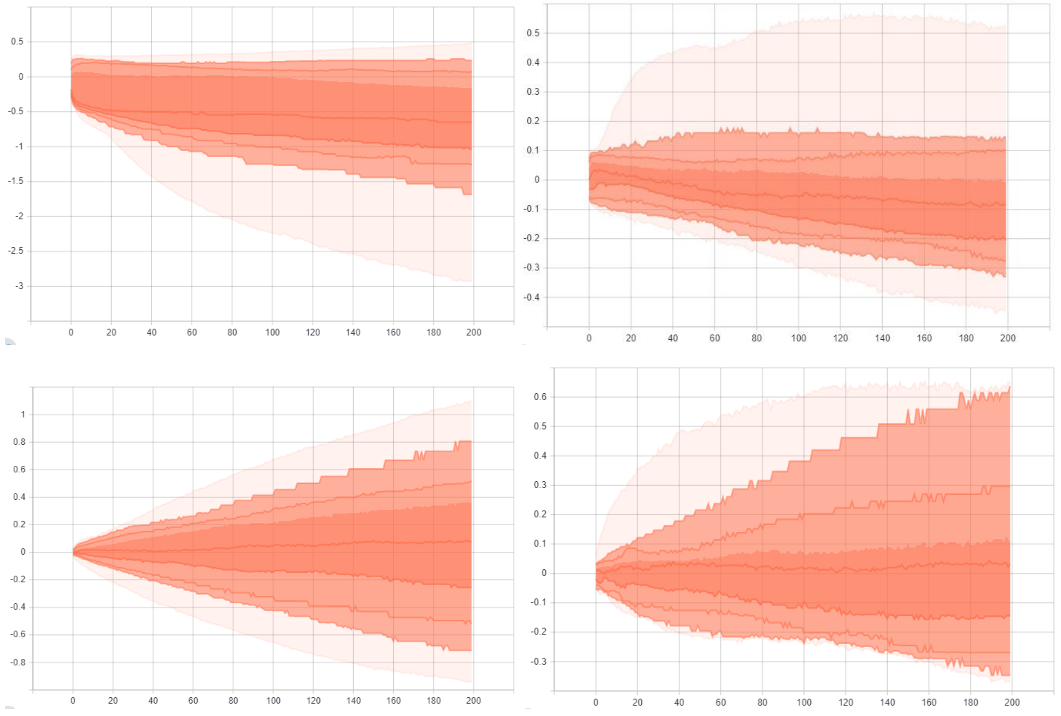
Wrist data set Distribution of bias weights for the four layers of the network using Sigmoid at 200 epochs with batch size 32. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



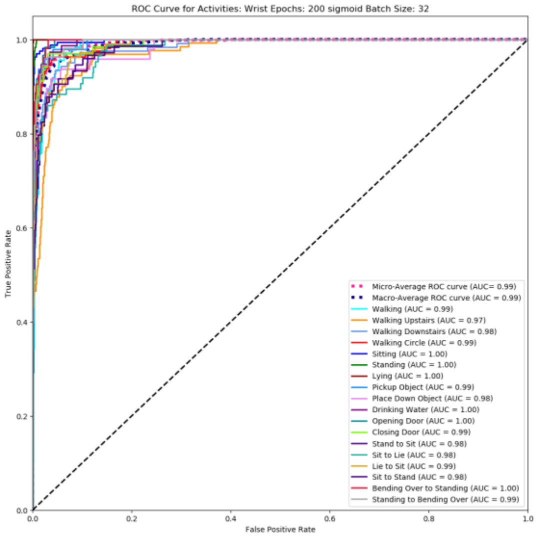
Waist data set Distribution of kernel weights for the four layers of the network using Sigmoid at 200 epochs with batch size 32. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.

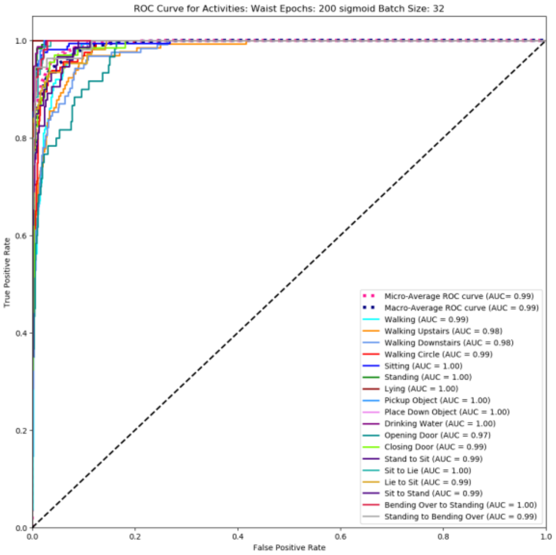


Waist data set Distribution of bias weights for the four layers of the network using Sigmoid at 200 epochs with batch size 32. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



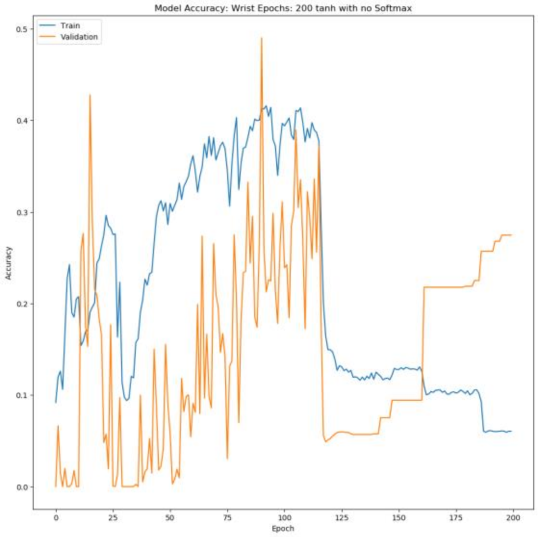
ROC Curve for wrist and waist

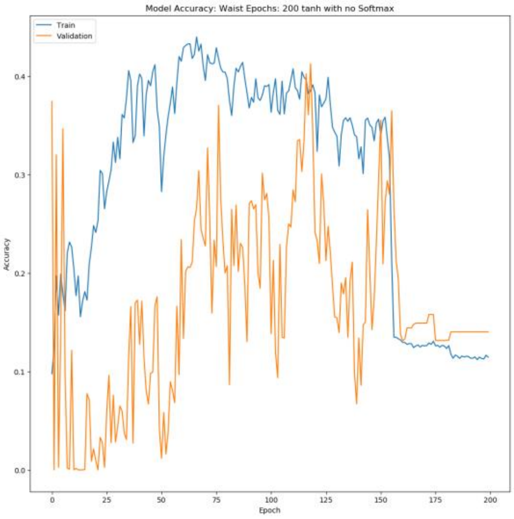




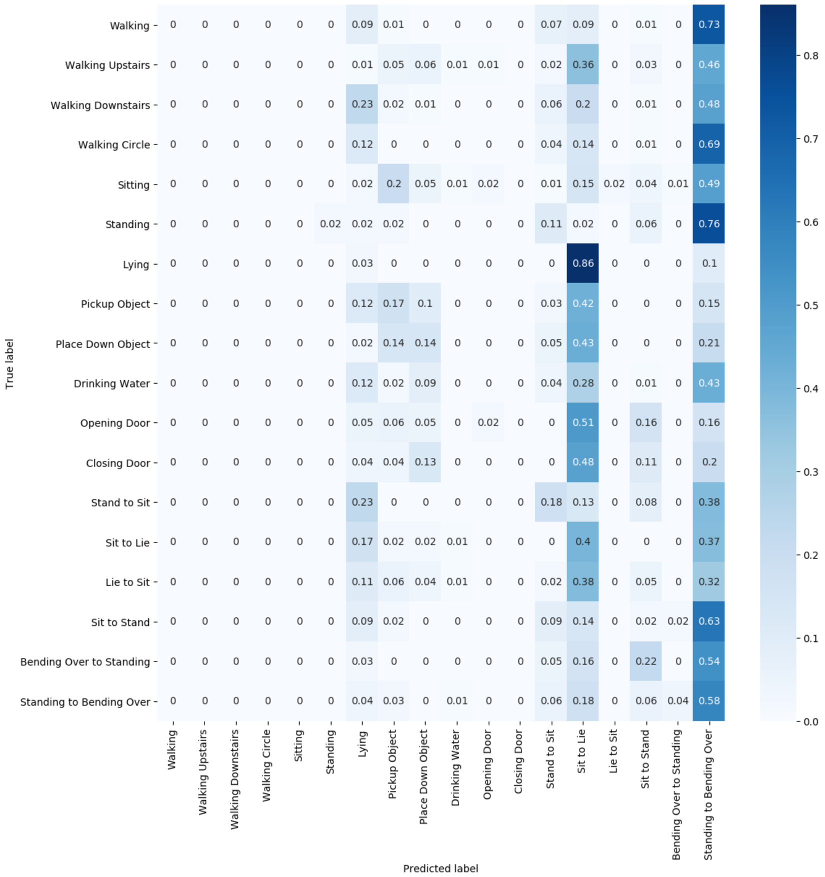
Appendix N – Additional Graphs: Tanh 200 Epochs No Softmax

Wrist and Waist Accuracy

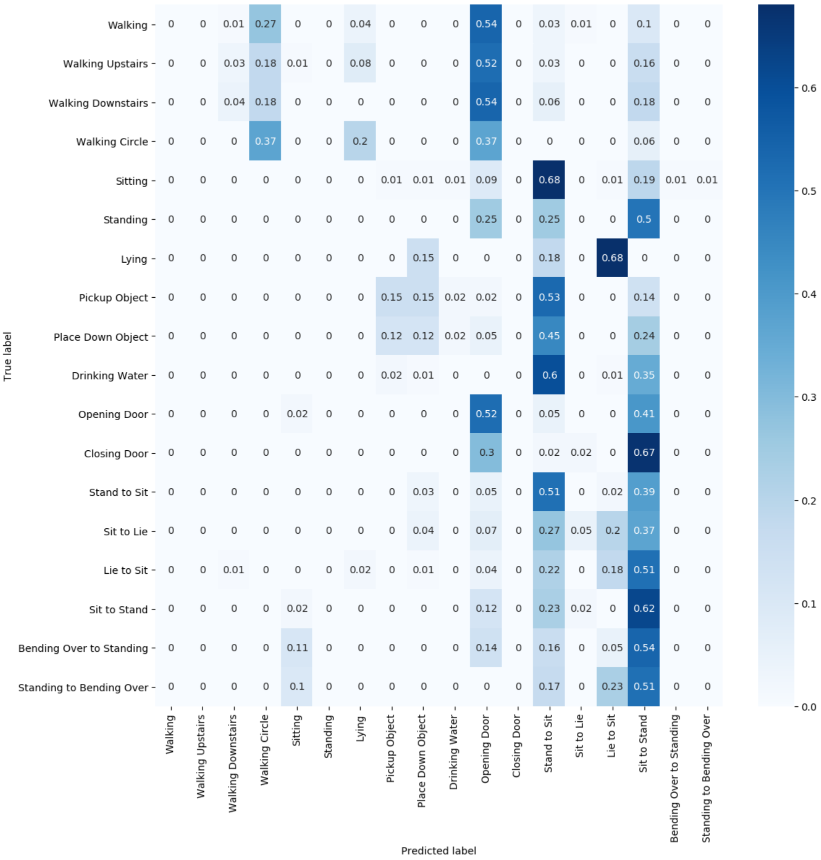




Wrist and Waist Confusion Matrix

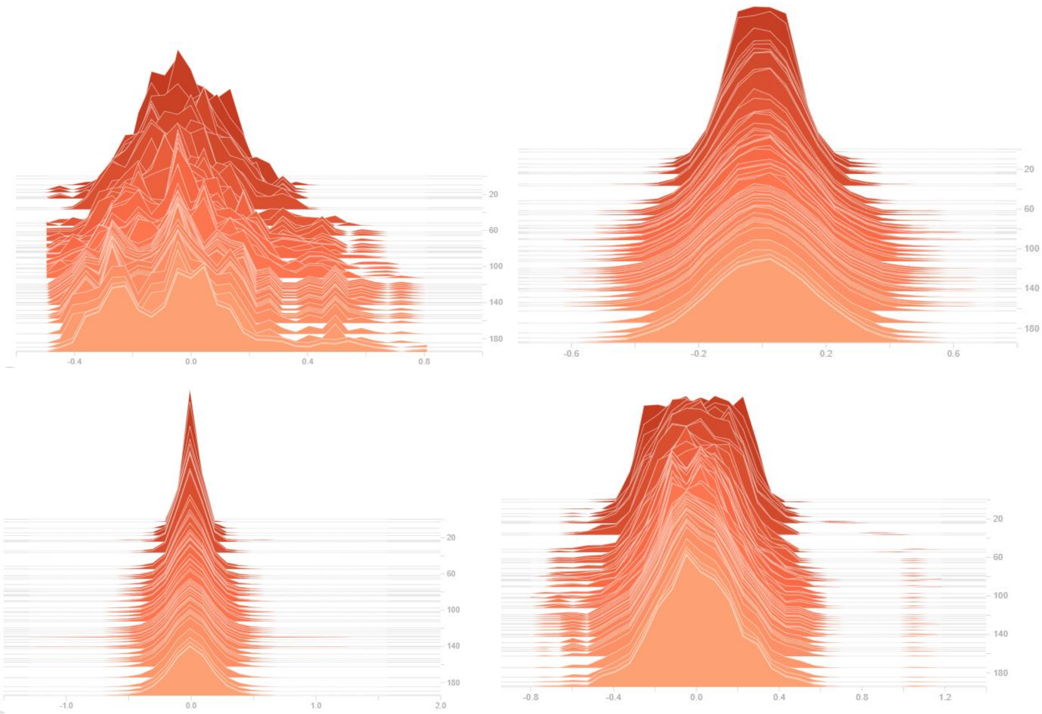


	Walking	Walking Upstairs	Walking Downstairs	Walking Circle	Sitting	Standing	Lying	Pickup Object	Place Down Object	Drinking Water	Opening Door	Closing Door	Stand to Sit	Sit to Lie	Lie to Sit	Sit to Stand	Bending Over to Standing	Standing to Bending Over
Walking	0	0	0	0	0	0	0.09	0.01	0	0	0	0	0.07	0.09	0	0.01	0	0.73
Walking Upstairs	0	0	0	0	0	0	0.01	0.05	0.06	0.01	0.01	0	0.02	0.36	0	0.03	0	0.46
Walking Downstairs	0	0	0	0	0	0	0.23	0.02	0.01	0	0	0	0.06	0.2	0	0.01	0	0.48
Walking Circle	0	0	0	0	0	0	0.12	0	0	0	0	0	0.04	0.14	0	0.01	0	0.69
Sitting	0	0	0	0	0	0	0.02	0.2	0.05	0.01	0.02	0	0.01	0.15	0.02	0.04	0.01	0.49
Standing	0	0	0	0	0	0.02	0.02	0.02	0	0	0	0	0.11	0.02	0	0.06	0	0.76
Lying	0	0	0	0	0	0	0.03	0	0	0	0	0	0	0.86	0	0	0	0.1
Pickup Object	0	0	0	0	0	0	0.12	0.17	0.1	0	0	0	0.03	0.42	0	0	0	0.15
Place Down Object	0	0	0	0	0	0	0.02	0.14	0.14	0	0	0	0.05	0.43	0	0	0	0.21
Drinking Water	0	0	0	0	0	0	0.12	0.02	0.09	0	0	0	0.04	0.28	0	0.01	0	0.43
Opening Door	0	0	0	0	0	0	0.05	0.06	0.05	0	0.02	0	0	0.51	0	0.16	0	0.16
Closing Door	0	0	0	0	0	0	0.04	0.04	0.13	0	0	0	0	0.48	0	0.11	0	0.2
Stand to Sit	0	0	0	0	0	0	0.23	0	0	0	0	0	0.18	0.13	0	0.08	0	0.38
Sit to Lie	0	0	0	0	0	0	0.17	0.02	0.02	0.01	0	0	0	0.4	0	0	0	0.37
Lie to Sit	0	0	0	0	0	0	0.11	0.06	0.04	0.01	0	0	0.02	0.38	0	0.05	0	0.32
Sit to Stand	0	0	0	0	0	0	0.09	0.02	0	0	0	0	0.09	0.14	0	0.02	0.02	0.63
Bending Over to Standing	0	0	0	0	0	0	0.03	0	0	0	0	0	0.05	0.16	0	0.22	0	0.54
Standing to Bending Over	0	0	0	0	0	0	0.04	0.03	0	0.01	0	0	0.06	0.18	0	0.06	0.04	0.58

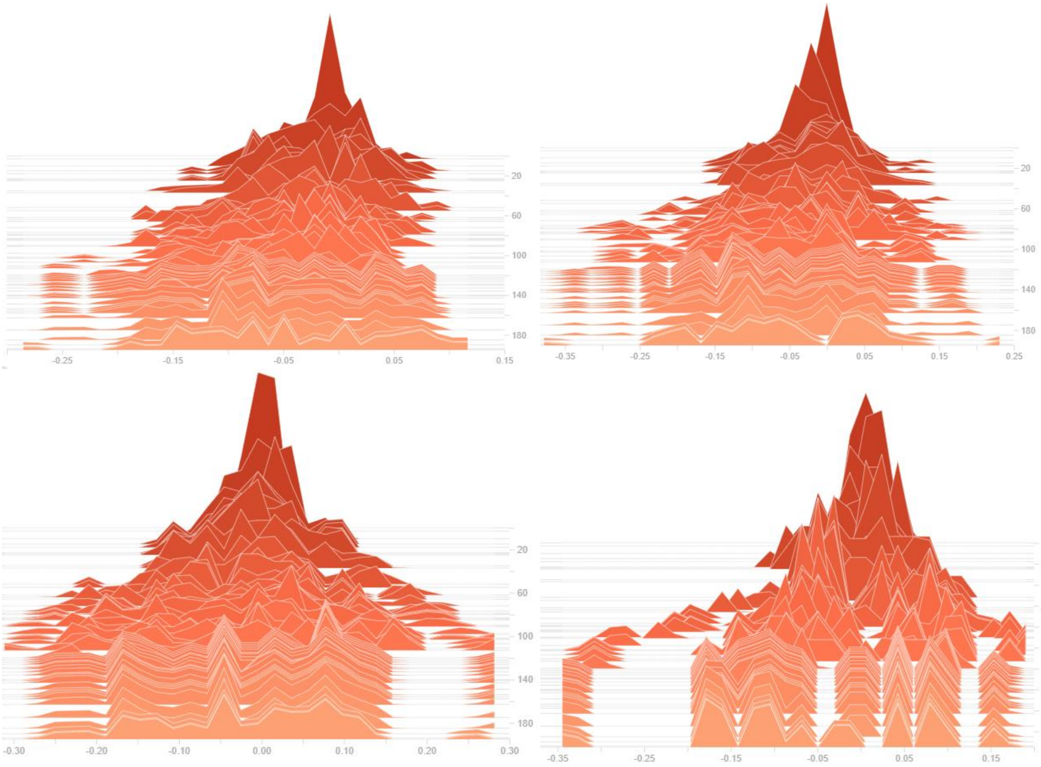


	Walking	Walking Upstairs	Walking Downstairs	Walking Circle	Sitting	Standing	Lying	Pickup Object	Place Down Object	Drinking Water	Opening Door	Closing Door	Stand to Sit	Sit to Lie	Lie to Sit	Sit to Stand	Bending Over to Standing	Standing to Bending Over
Walking	0	0	0.01	0.27	0	0	0.04	0	0	0	0.54	0	0.03	0.01	0	0.1	0	0
Walking Upstairs	0	0	0.03	0.18	0.01	0	0.08	0	0	0	0.52	0	0.03	0	0	0.16	0	0
Walking Downstairs	0	0	0.04	0.18	0	0	0	0	0	0	0.54	0	0.06	0	0	0.18	0	0
Walking Circle	0	0	0	0.37	0	0	0.2	0	0	0	0.37	0	0	0	0	0.06	0	0
Sitting	0	0	0	0	0	0	0	0.01	0.01	0.01	0.09	0	0.68	0	0.01	0.19	0.01	0.01
Standing	0	0	0	0	0	0	0	0	0	0	0.25	0	0.25	0	0	0.5	0	0
Lying	0	0	0	0	0	0	0	0	0.15	0	0	0	0.18	0	0.68	0	0	0
Pickup Object	0	0	0	0	0	0	0	0.15	0.15	0.02	0.02	0	0.53	0	0	0.14	0	0
Place Down Object	0	0	0	0	0	0	0	0.12	0.12	0.02	0.05	0	0.45	0	0	0.24	0	0
Drinking Water	0	0	0	0	0	0	0	0.02	0.01	0	0	0	0.6	0	0.01	0.35	0	0
Opening Door	0	0	0	0	0.02	0	0	0	0	0	0.52	0	0.05	0	0	0.41	0	0
Closing Door	0	0	0	0	0	0	0	0	0	0	0.3	0	0.02	0.02	0	0.67	0	0
Stand to Sit	0	0	0	0	0	0	0	0	0.03	0	0.05	0	0.51	0	0.02	0.39	0	0
Sit to Lie	0	0	0	0	0	0	0	0	0.04	0	0.07	0	0.27	0.05	0.2	0.37	0	0
Lie to Sit	0	0	0.01	0	0	0	0.02	0	0.01	0	0.04	0	0.22	0	0.18	0.51	0	0
Sit to Stand	0	0	0	0	0.02	0	0	0	0	0	0.12	0	0.23	0.02	0	0.62	0	0
Bending Over to Standing	0	0	0	0	0.11	0	0	0	0	0	0.14	0	0.16	0	0.05	0.54	0	0
Standing to Bending Over	0	0	0	0	0.1	0	0	0	0	0	0	0	0.17	0	0.23	0.51	0	0

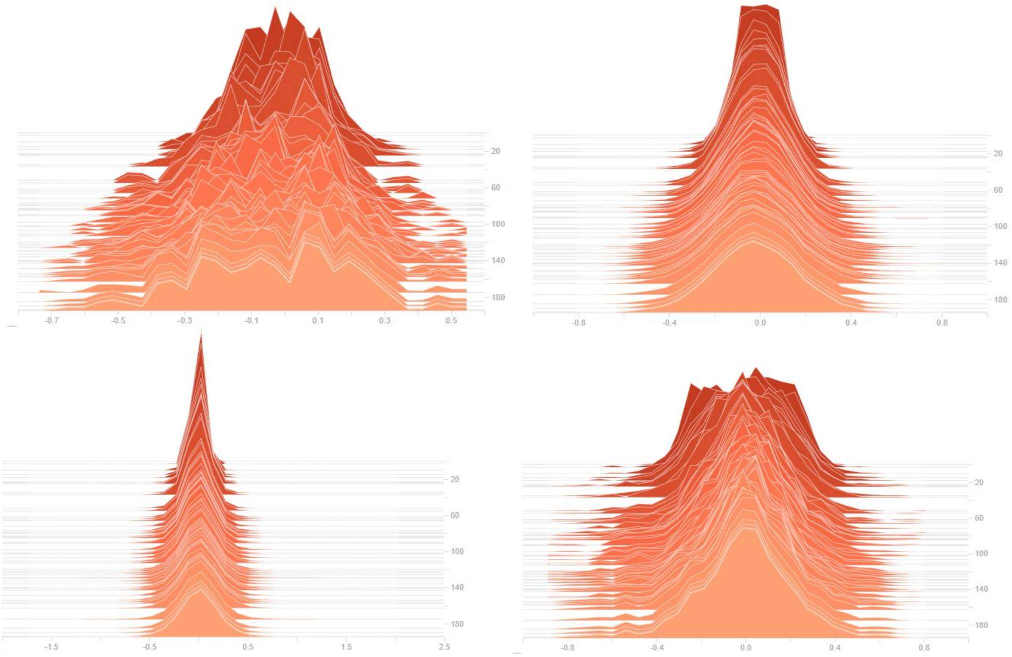
Wrist data set Histogram of kernel weights for the four layers of the network using Tanh at 200 epochs with no Softmax. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



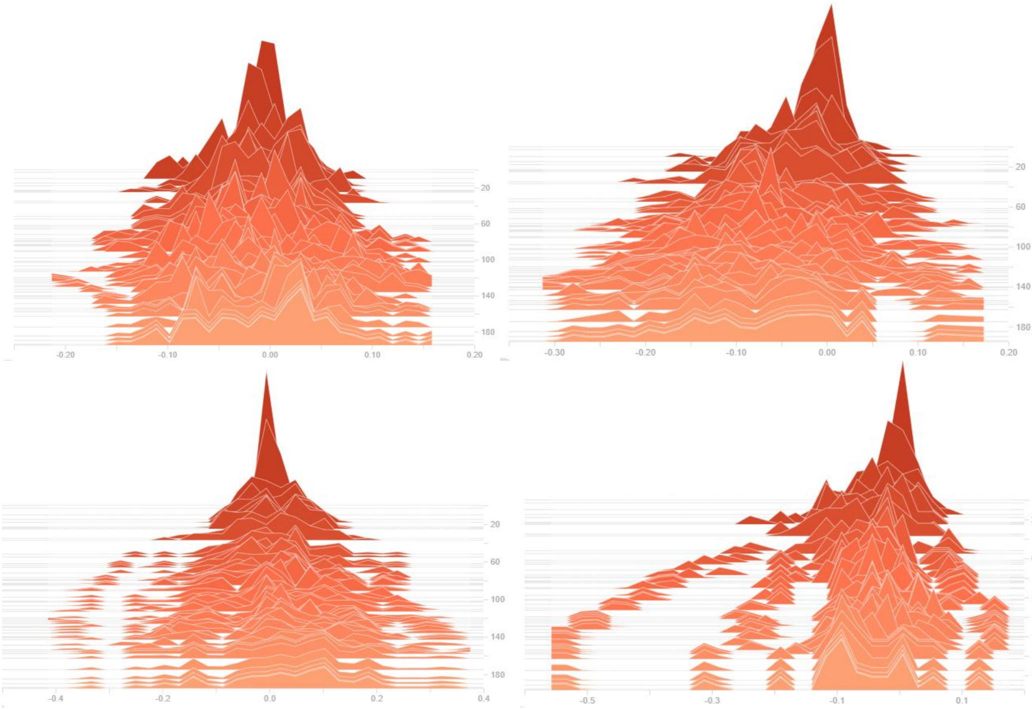
Wrist data set Histogram of bias weights for the four layers of the network using Tanh at 200 epochs with no Softmax. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



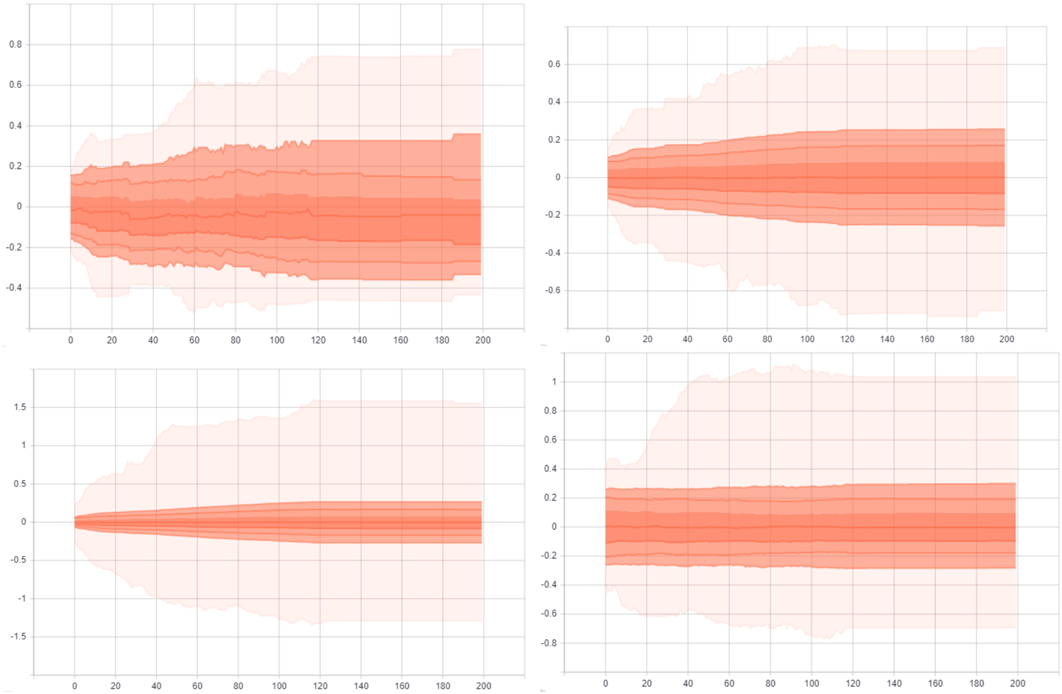
Waist data set Histogram of kernel weights for the four layers of the network using Tanh at 200 epochs with no Softmax. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



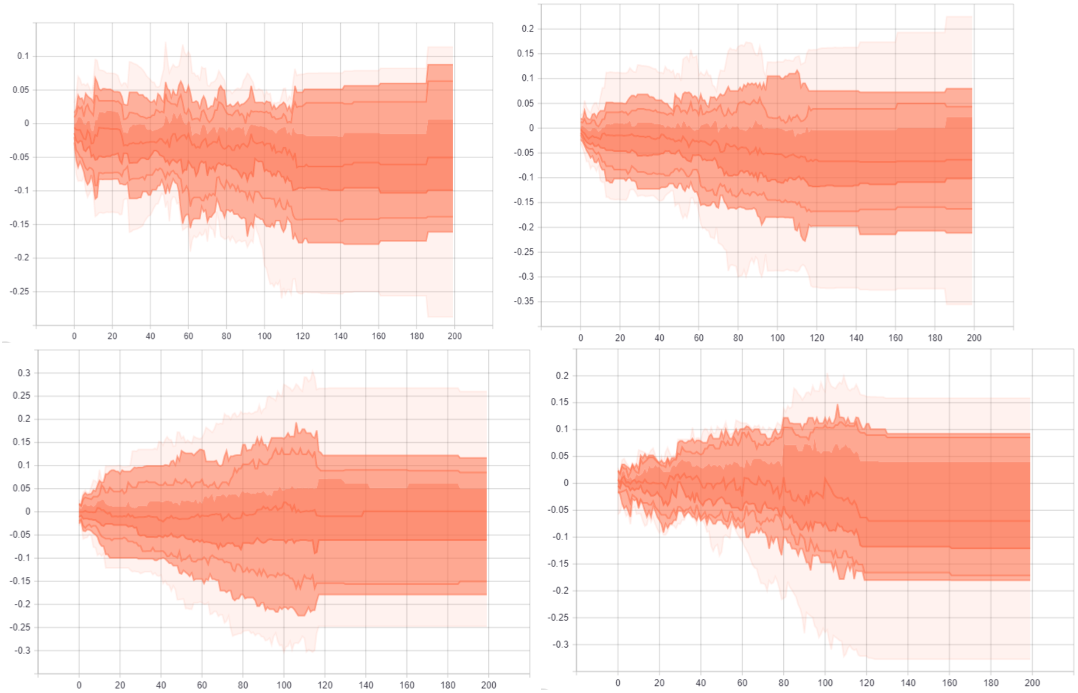
Waist data set Histogram of bias weights for the four layers of the network using Tanh at 200 epochs with no Softmax. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



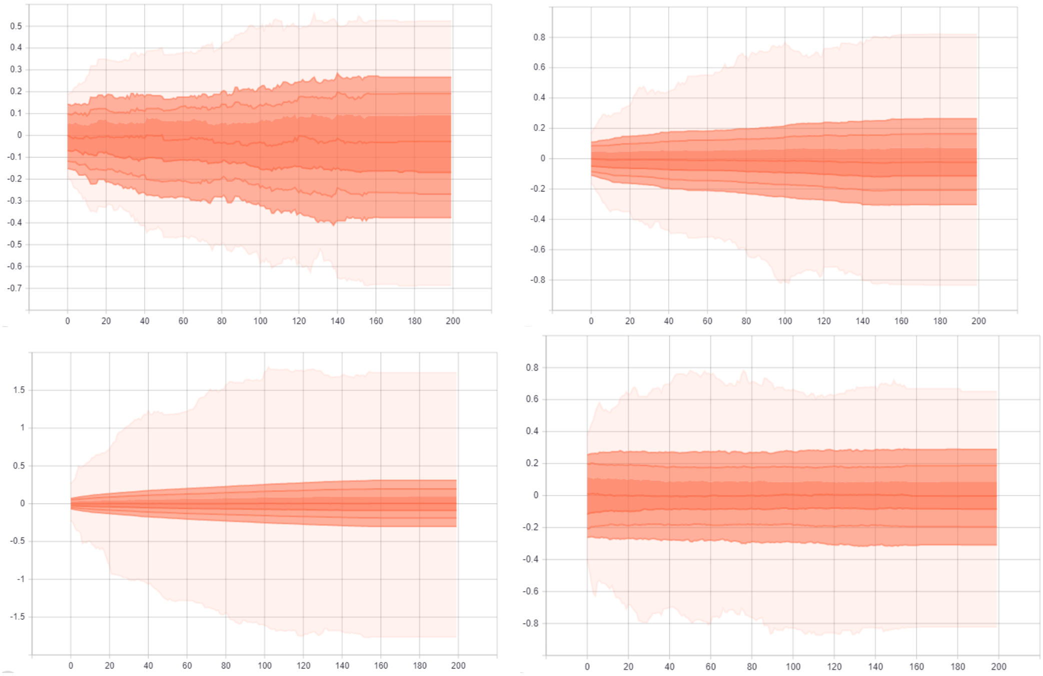
Wrist data set Distribution of kernel weights for the four layers of the network using Tanh at 200 epochs with no Softmax. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



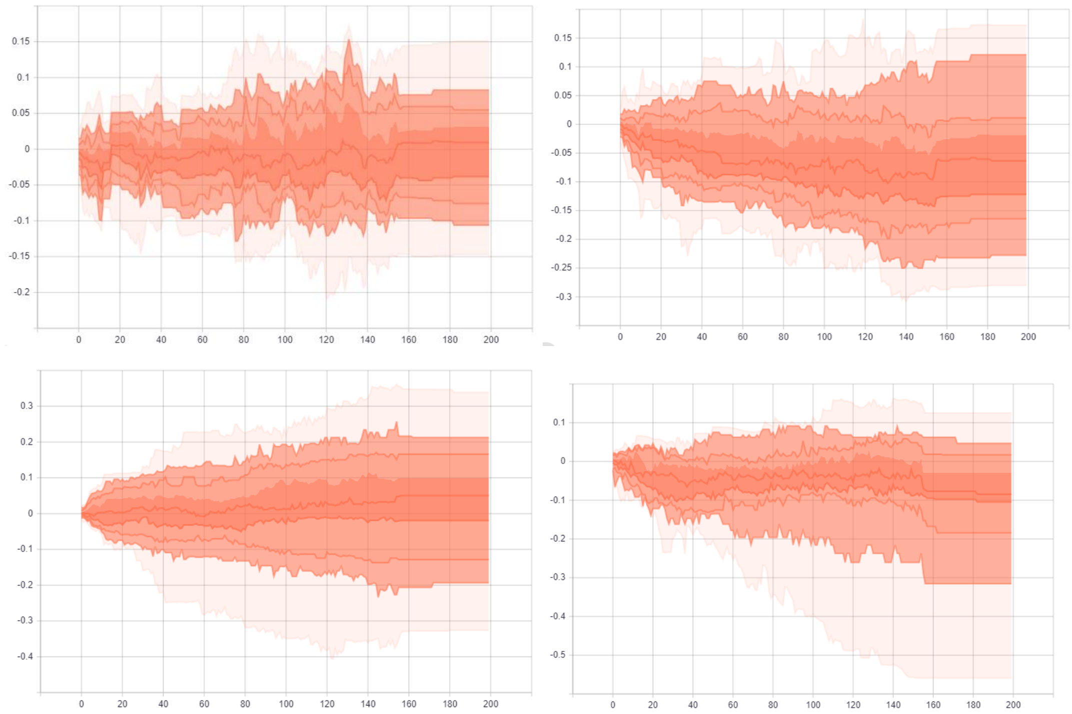
Wrist data set Distribution of bias weights for the four layers of the network using Tanh at 200 epochs with no Softmax. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



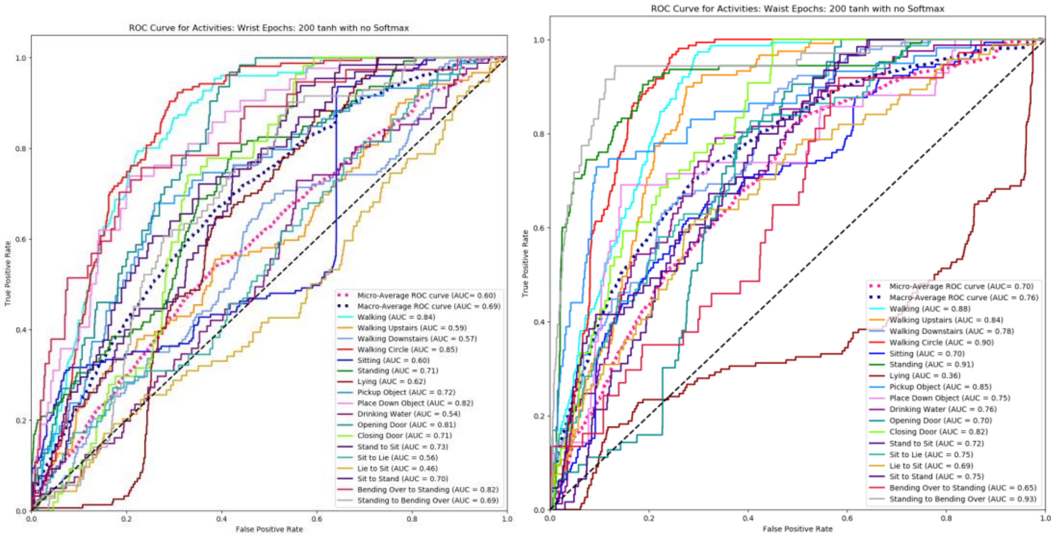
Waist data set Distribution of kernel weights for the four layers of the network using Tanh at 200 epochs with no Softmax. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



Waist data set Distribution of bias weights for the four layers of the network using Tanh at 200 epochs with no Softmax. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.

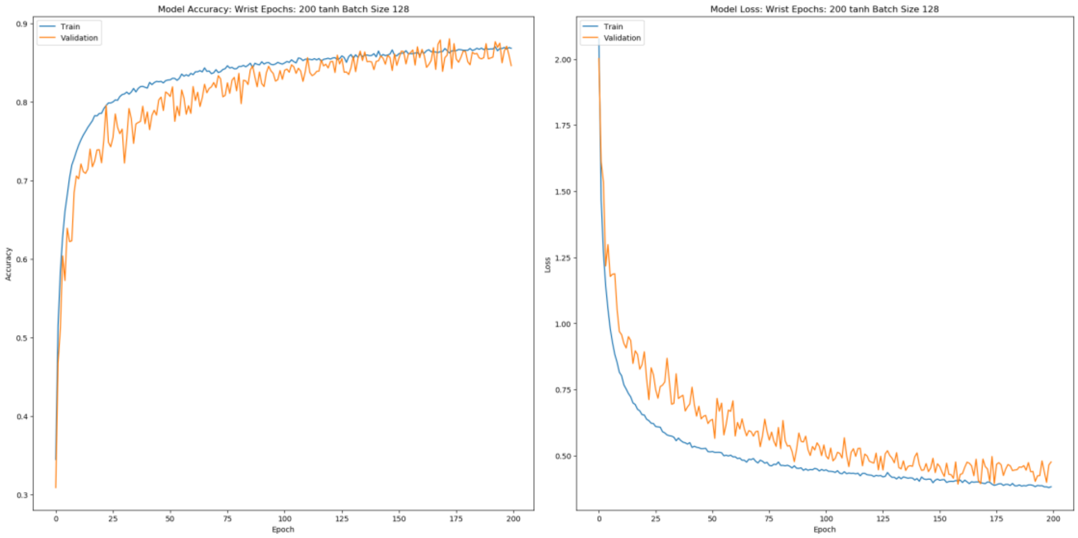


ROC Curve for Wrist and Waist

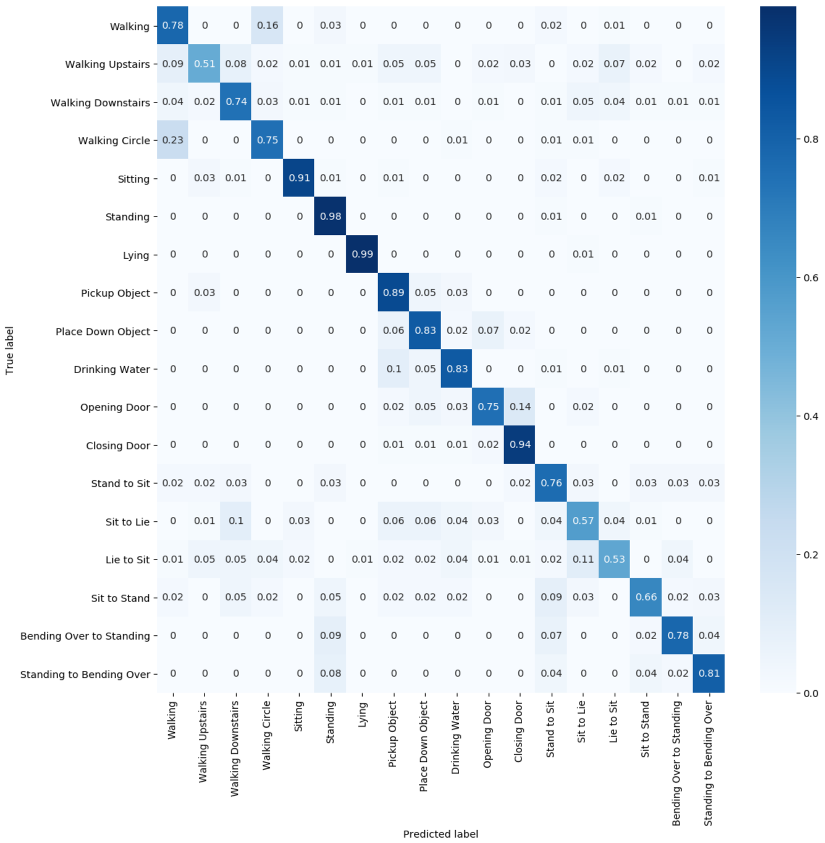


Appendix O – Additional Graphs: Tanh 200 Epochs Batch Size 128

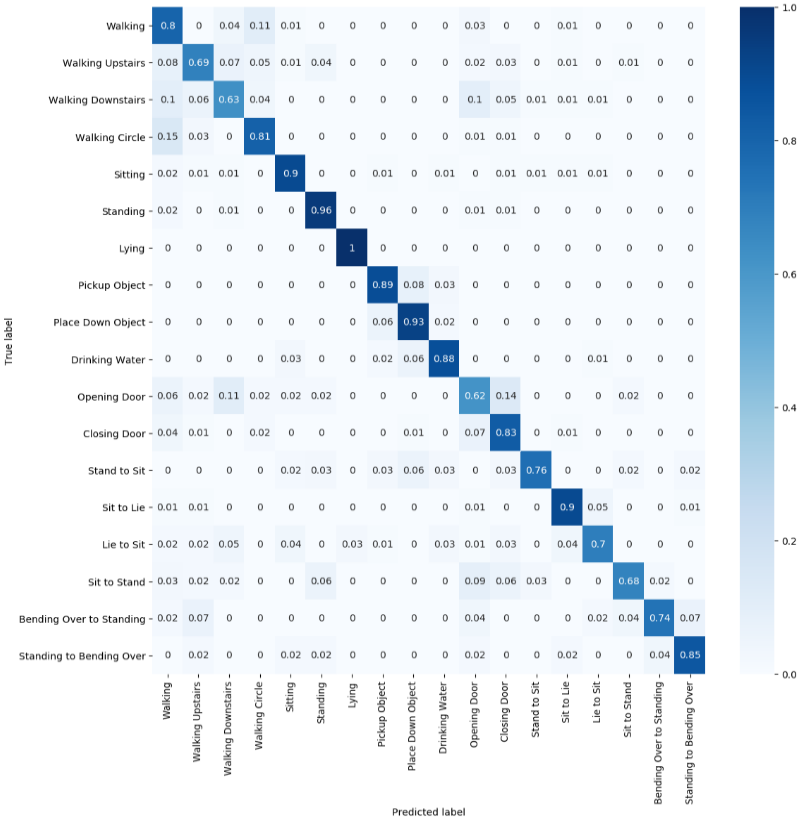
Wrist accuracy and loss



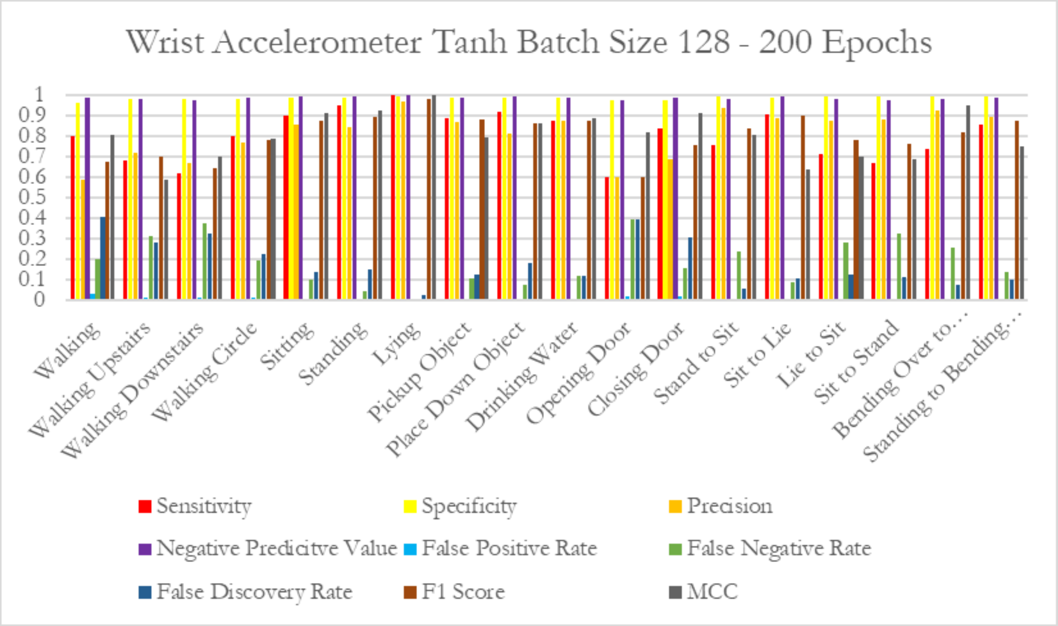
Wrist and waist confusion matrix

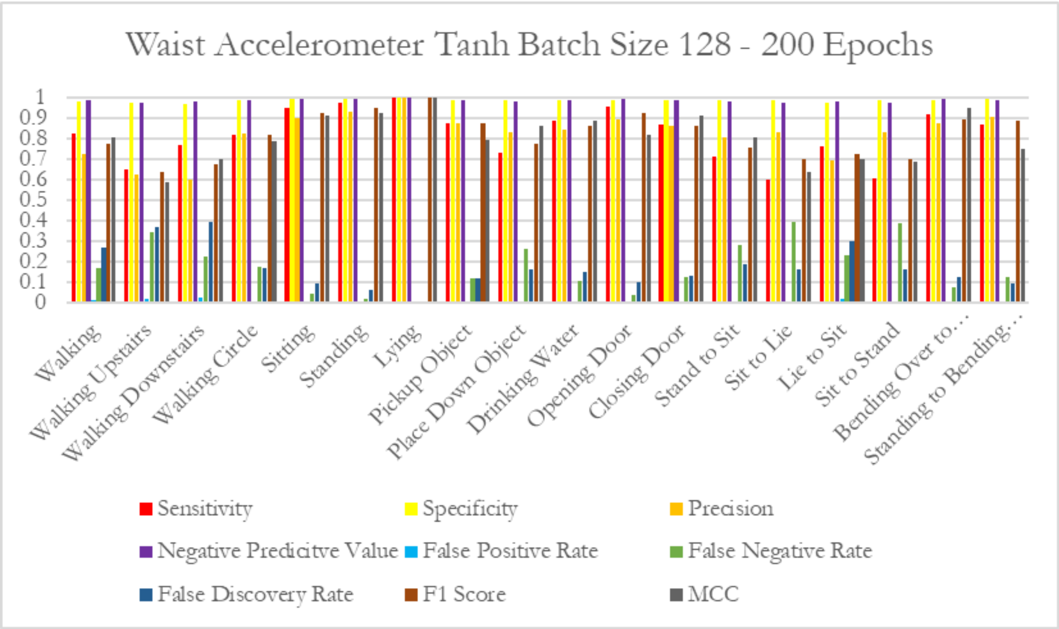


	Walking	Walking Upstairs	Walking Downstairs	Walking Circle	Sitting	Standing	Lying	Pickup Object	Place Down Object	Drinking Water	Opening Door	Closing Door	Stand to Sit	Sit to Lie	Lie to Sit	Sit to Stand	Bending Over to Standing	Standing to Bending Over
Walking	0.78	0	0	0.16	0	0.03	0	0	0	0	0	0	0.02	0	0.01	0	0	0
Walking Upstairs	0.09	0.51	0.08	0.02	0.01	0.01	0.01	0.05	0.05	0	0.02	0.03	0	0.02	0.07	0.02	0	0.02
Walking Downstairs	0.04	0.02	0.74	0.03	0.01	0.01	0	0.01	0.01	0	0.01	0	0.01	0.05	0.04	0.01	0.01	0.01
Walking Circle	0.23	0	0	0.75	0	0	0	0	0	0.01	0	0	0.01	0.01	0	0	0	0
Sitting	0	0.03	0.01	0	0.91	0.01	0	0.01	0	0	0	0	0.02	0	0.02	0	0	0.01
Standing	0	0	0	0	0	0.98	0	0	0	0	0	0	0.01	0	0	0.01	0	0
Lying	0	0	0	0	0	0	0.99	0	0	0	0	0	0	0.01	0	0	0	0
Pickup Object	0	0.03	0	0	0	0	0	0.89	0.05	0.03	0	0	0	0	0	0	0	0
Place Down Object	0	0	0	0	0	0	0	0.06	0.83	0.02	0.07	0.02	0	0	0	0	0	0
Drinking Water	0	0	0	0	0	0	0	0.1	0.05	0.83	0	0	0.01	0	0.01	0	0	0
Opening Door	0	0	0	0	0	0	0	0.02	0.05	0.03	0.75	0.14	0	0.02	0	0	0	0
Closing Door	0	0	0	0	0	0	0	0.01	0.01	0.01	0.02	0.94	0	0	0	0	0	0
Stand to Sit	0.02	0.02	0.03	0	0	0.03	0	0	0	0	0	0.02	0.76	0.03	0	0.03	0.03	0.03
Sit to Lie	0	0.01	0.1	0	0.03	0	0	0.06	0.06	0.04	0.03	0	0.04	0.57	0.04	0.01	0	0
Lie to Sit	0.01	0.05	0.05	0.04	0.02	0	0.01	0.02	0.02	0.04	0.01	0.01	0.02	0.11	0.53	0	0.04	0
Sit to Stand	0.02	0	0.05	0.02	0	0.05	0	0.02	0.02	0.02	0	0	0.09	0.03	0	0.66	0.02	0.03
Bending Over to Standing	0	0	0	0	0	0.09	0	0	0	0	0	0	0.07	0	0	0.02	0.78	0.04
Standing to Bending Over	0	0	0	0	0	0.08	0	0	0	0	0	0	0.04	0	0	0.04	0.02	0.81

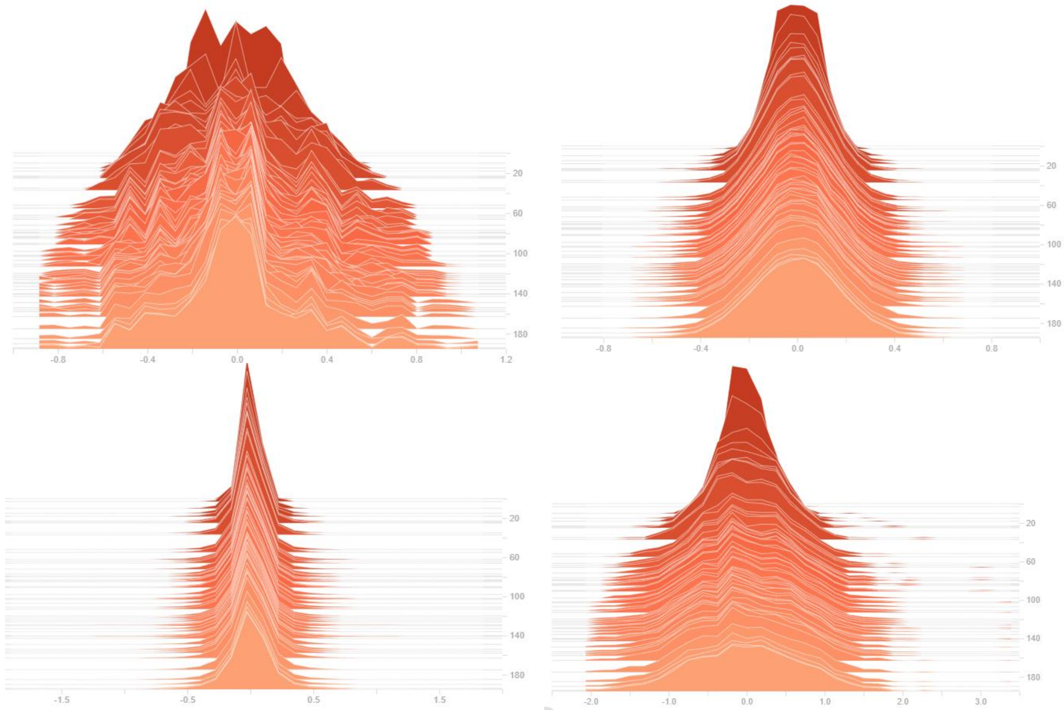


	Walking	Walking Upstairs	Walking Downstairs	Walking Circle	Sitting	Standing	Lying	Pickup Object	Place Down Object	Drinking Water	Opening Door	Closing Door	Stand to Sit	Sit to Lie	Lie to Sit	Sit to Stand	Bending Over to Standing	Standing to Bending Over
Walking	0.8	0	0.04	0.11	0.01	0	0	0	0	0	0.03	0	0	0.01	0	0	0	0
Walking Upstairs	0.08	0.69	0.07	0.05	0.01	0.04	0	0	0	0	0.02	0.03	0	0.01	0	0.01	0	0
Walking Downstairs	0.1	0.06	0.63	0.04	0	0	0	0	0	0	0.1	0.05	0.01	0.01	0.01	0	0	0
Walking Circle	0.15	0.03	0	0.81	0	0	0	0	0	0	0.01	0.01	0	0	0	0	0	0
Sitting	0.02	0.01	0.01	0	0.9	0	0	0.01	0	0.01	0	0.01	0.01	0.01	0.01	0	0	0
Standing	0.02	0	0.01	0	0	0.96	0	0	0	0	0.01	0.01	0	0	0	0	0	0
Lying	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Pickup Object	0	0	0	0	0	0	0	0.89	0.08	0.03	0	0	0	0	0	0	0	0
Place Down Object	0	0	0	0	0	0	0	0.06	0.93	0.02	0	0	0	0	0	0	0	0
Drinking Water	0	0	0	0	0.03	0	0	0.02	0.06	0.88	0	0	0	0	0.01	0	0	0
Opening Door	0.06	0.02	0.11	0.02	0.02	0.02	0	0	0	0	0.62	0.14	0	0	0	0.02	0	0
Closing Door	0.04	0.01	0	0.02	0	0	0	0	0.01	0	0.07	0.83	0	0.01	0	0	0	0
Stand to Sit	0	0	0	0	0.02	0.03	0	0.03	0.06	0.03	0	0.03	0.76	0	0	0.02	0	0.02
Sit to Lie	0.01	0.01	0	0	0	0	0	0	0	0	0.01	0	0	0.9	0.05	0	0	0.01
Lie to Sit	0.02	0.02	0.05	0	0.04	0	0.03	0.01	0	0.03	0.01	0.03	0	0.04	0.7	0	0	0
Sit to Stand	0.03	0.02	0.02	0	0	0.06	0	0	0	0	0.09	0.06	0.03	0	0	0.68	0.02	0
Bending Over to Standing	0.02	0.07	0	0	0	0	0	0	0	0	0.04	0	0	0	0.02	0.04	0.74	0.07
Standing to Bending Over	0	0.02	0	0	0.02	0.02	0	0	0	0	0.02	0	0	0.02	0	0	0.04	0.85

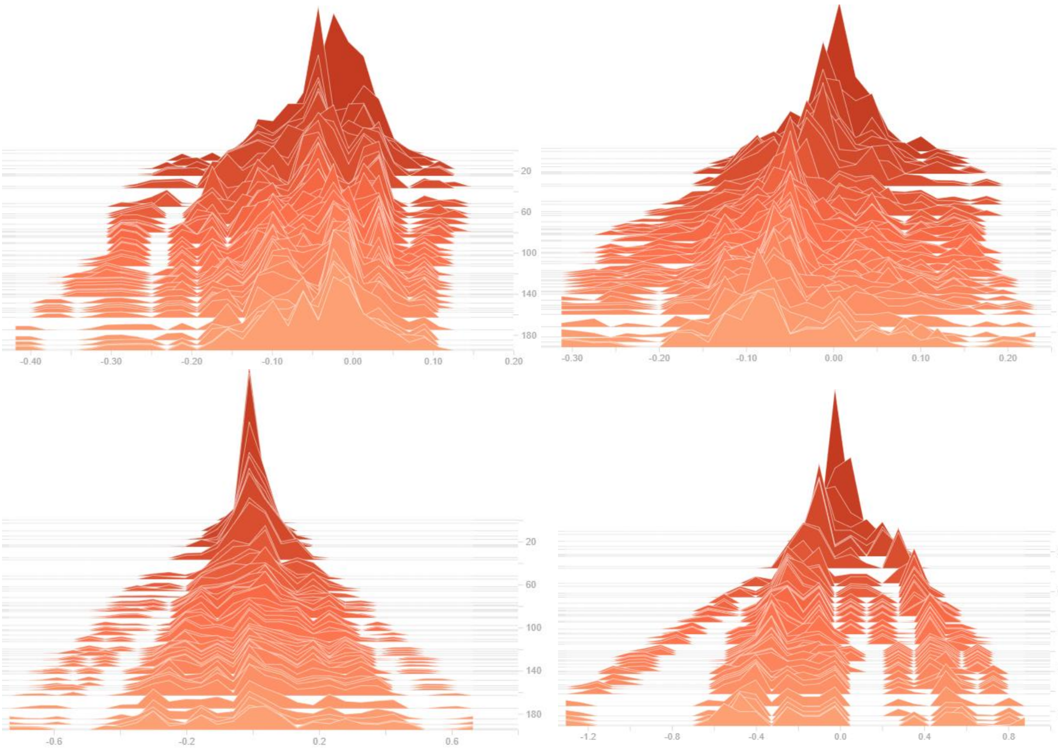




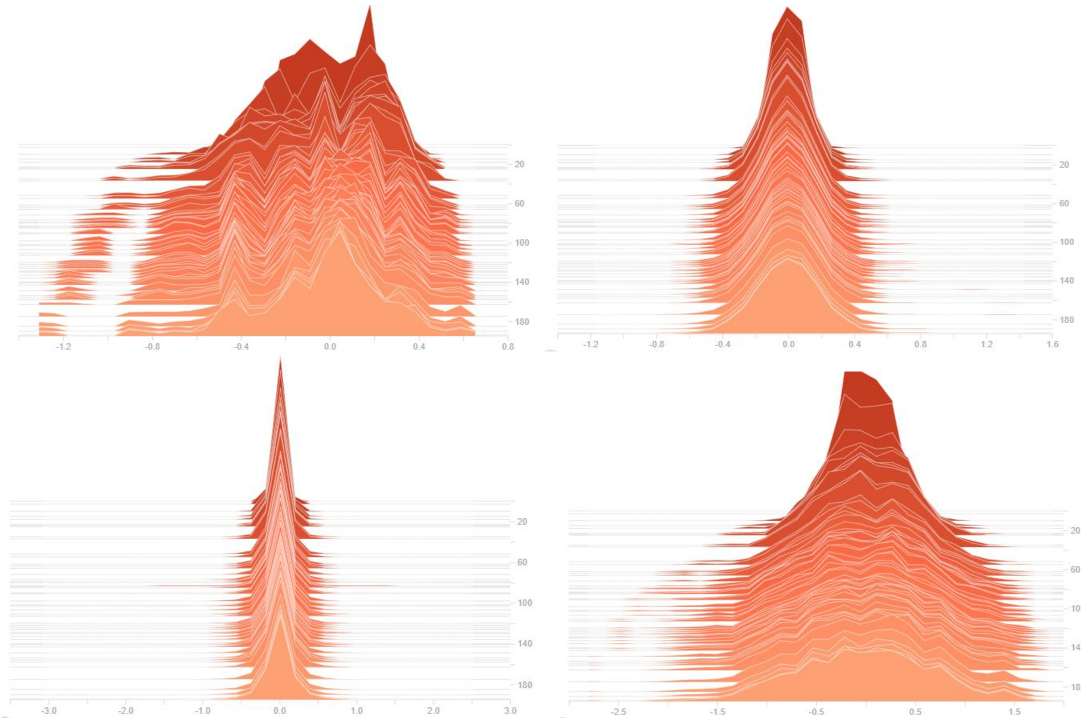
Wrist data set Histogram of kernel weights for the four layers of the network using Tanh at 200 epochs with batch size 128. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



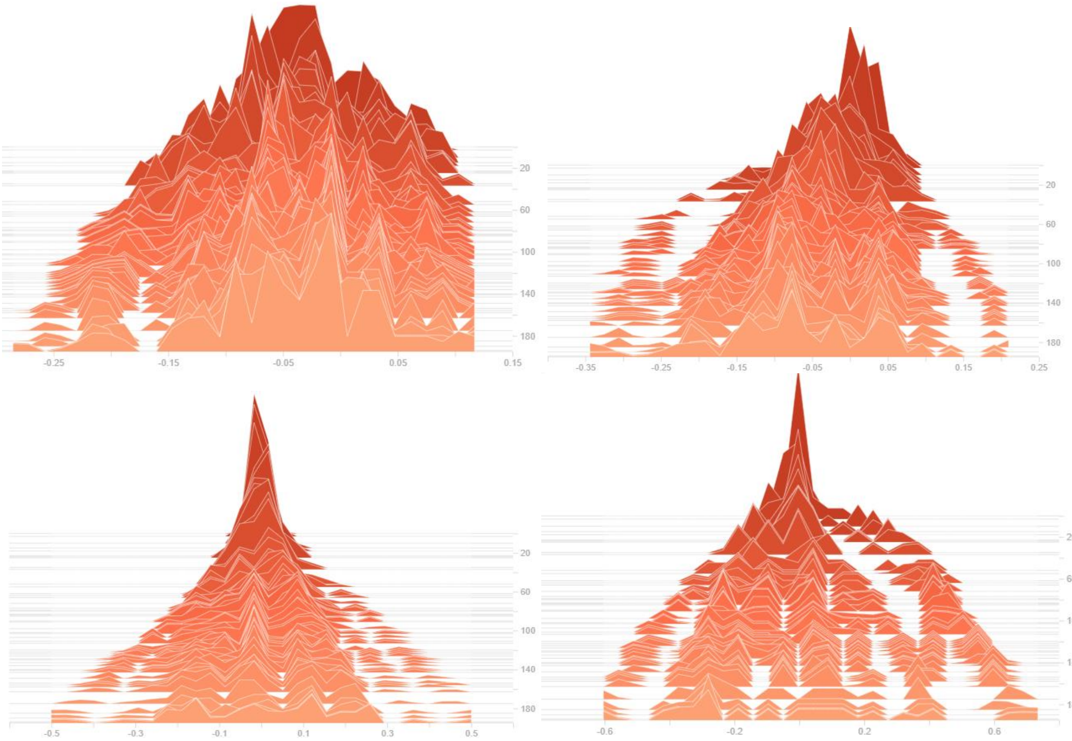
Wrist data set Histogram of bias weights for the four layers of the network using Tanh at 200 epochs with batch size 128. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



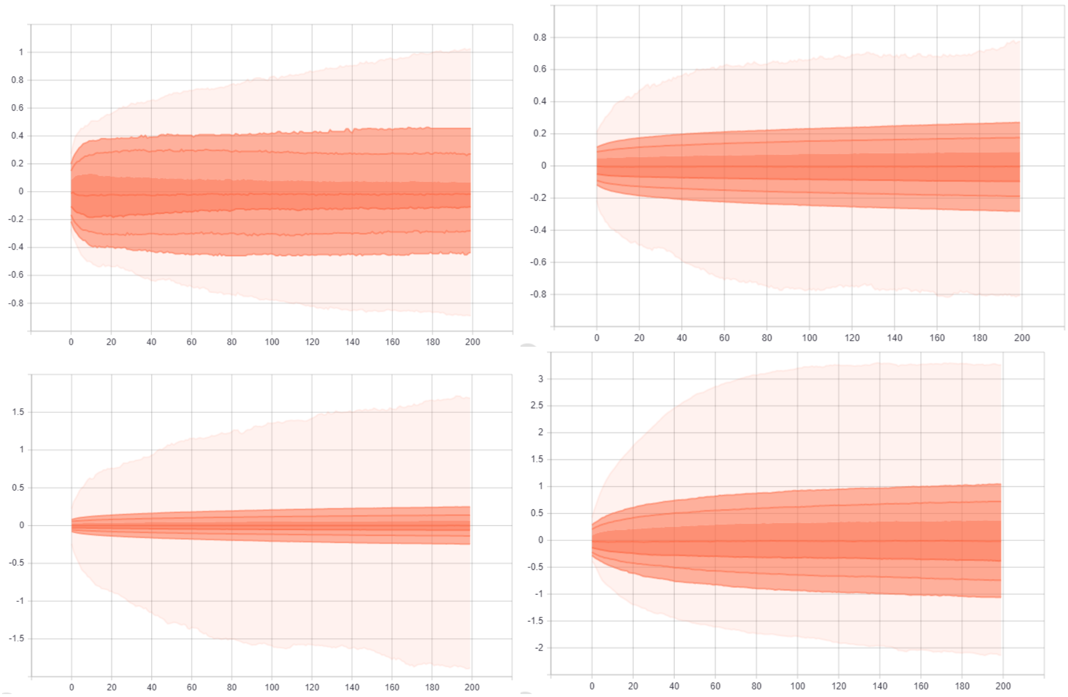
Waist data set Histogram of kernel weights for the four layers of the network using Tanh at 200 epochs with batch size 128. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



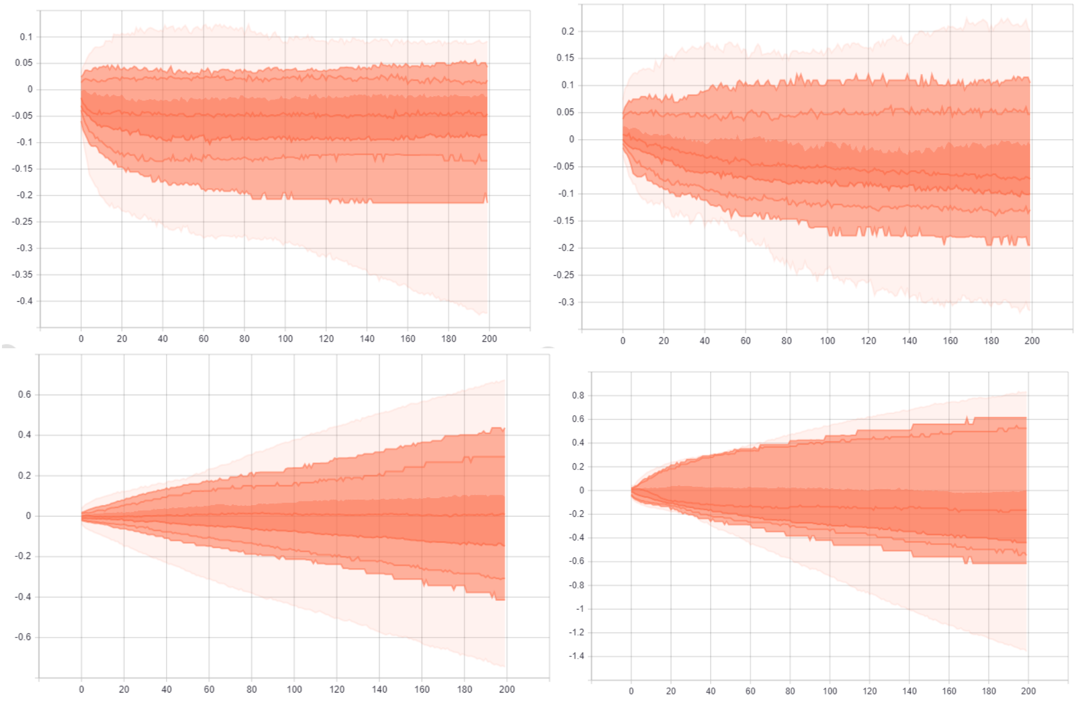
Waist data set Histogram of bias weights for the four layers of the network using Tanh at 200 epochs with batch size 128. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



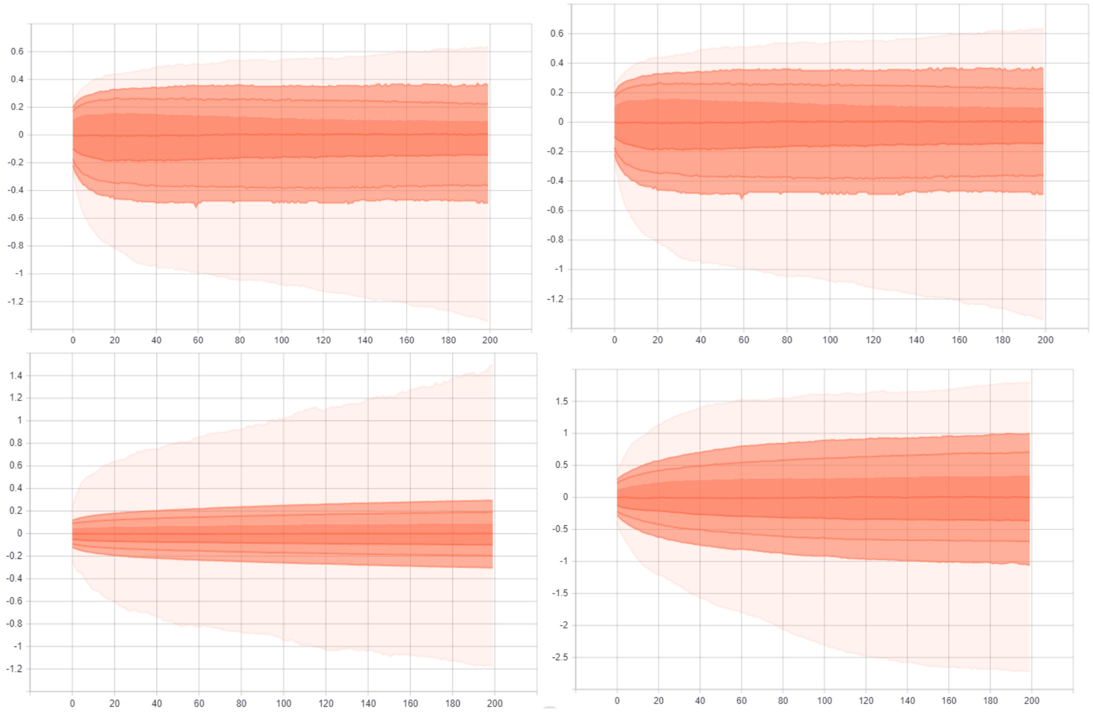
Wrist data set Distribution of kernel weights for the four layers of the network using Tanh at 200 epochs with batch size 128. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



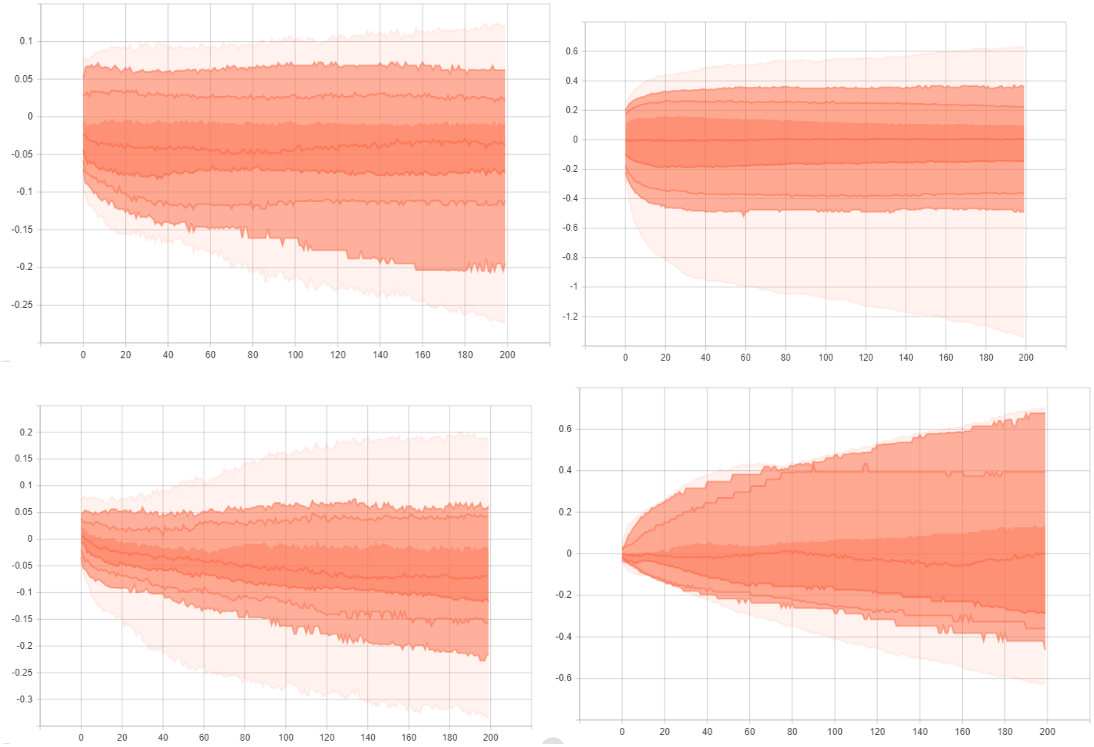
Wrist data set Distribution of bias weights for the four layers of the network using Tanh at 200 epochs with batch size 128. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



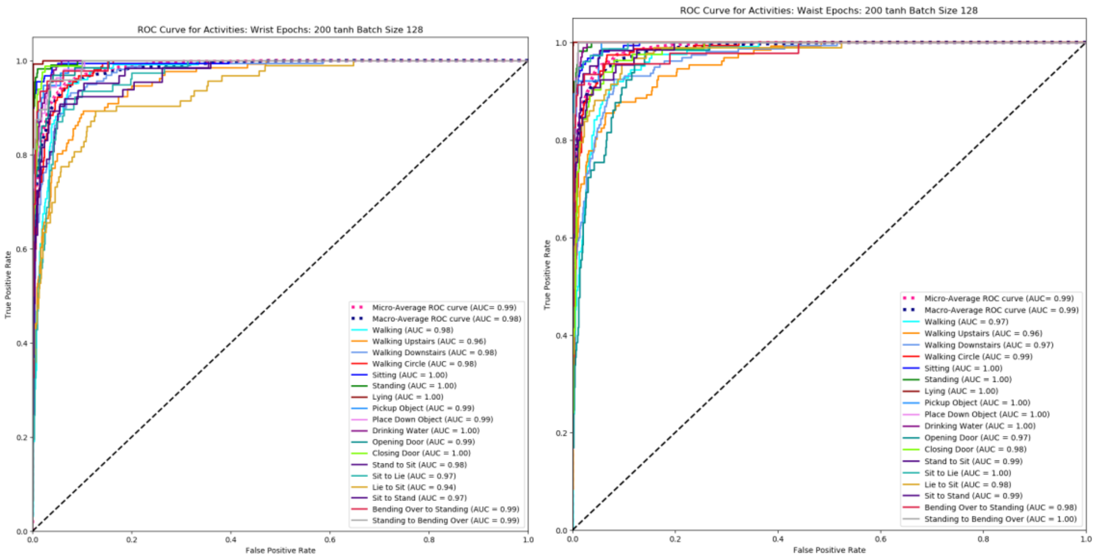
Waist data set Distribution of kernel weights for the four layers of the network using Tanh at 200 epochs with batch size 128. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



Waist data set Distribution of bias weights for the four layers of the network using Tanh at 200 epochs with batch size 128. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.

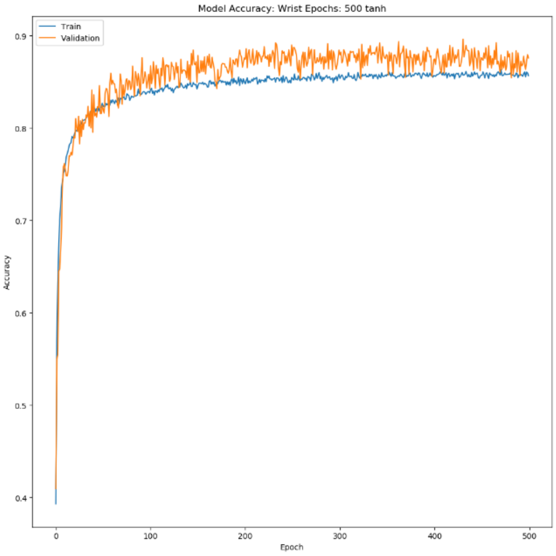


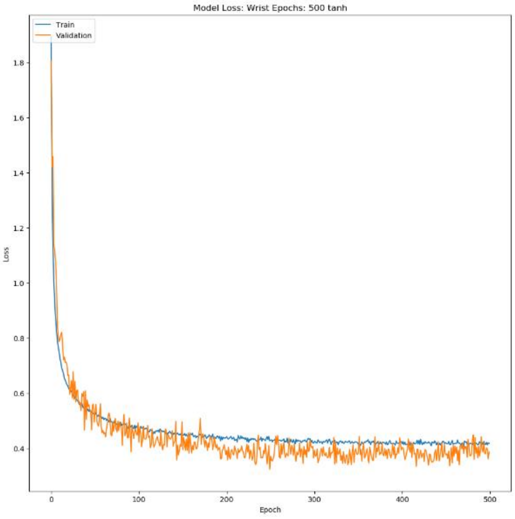
ROC Curve for Wrist and waist



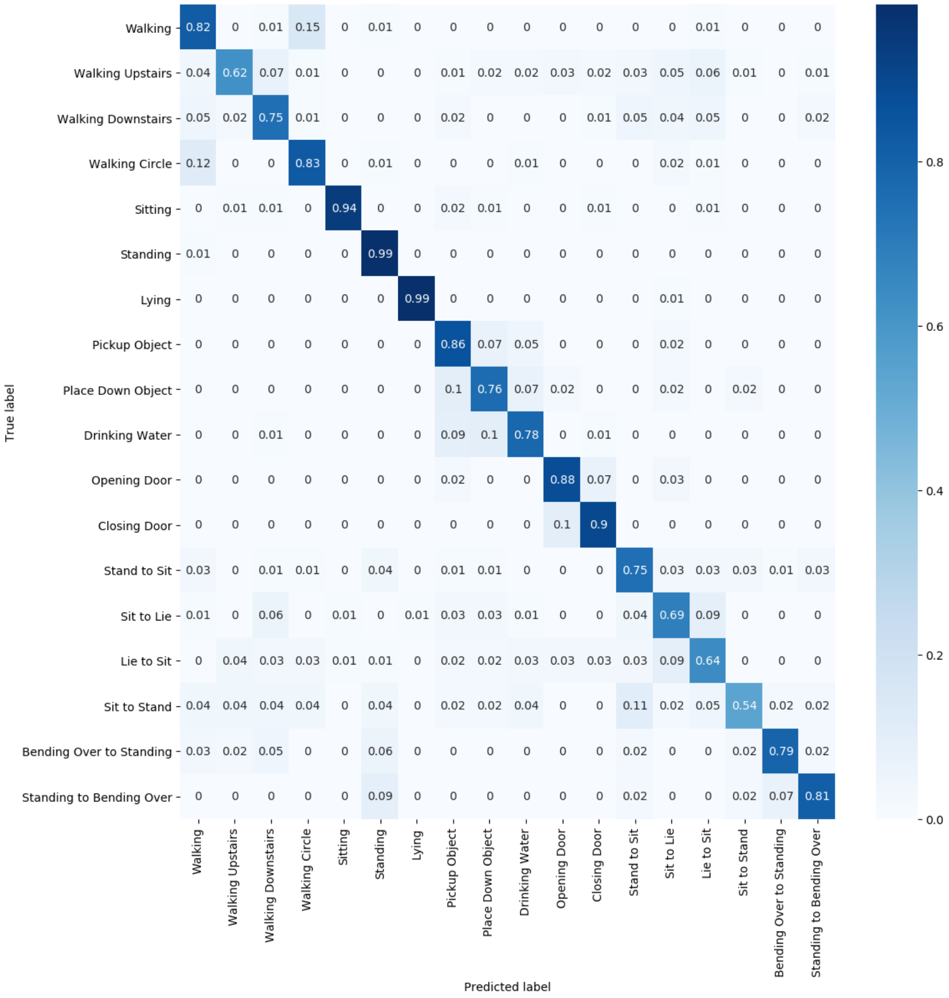
Appendix P – Additional Graphs: Tanh 500 Epochs

Wrist accuracy and loss



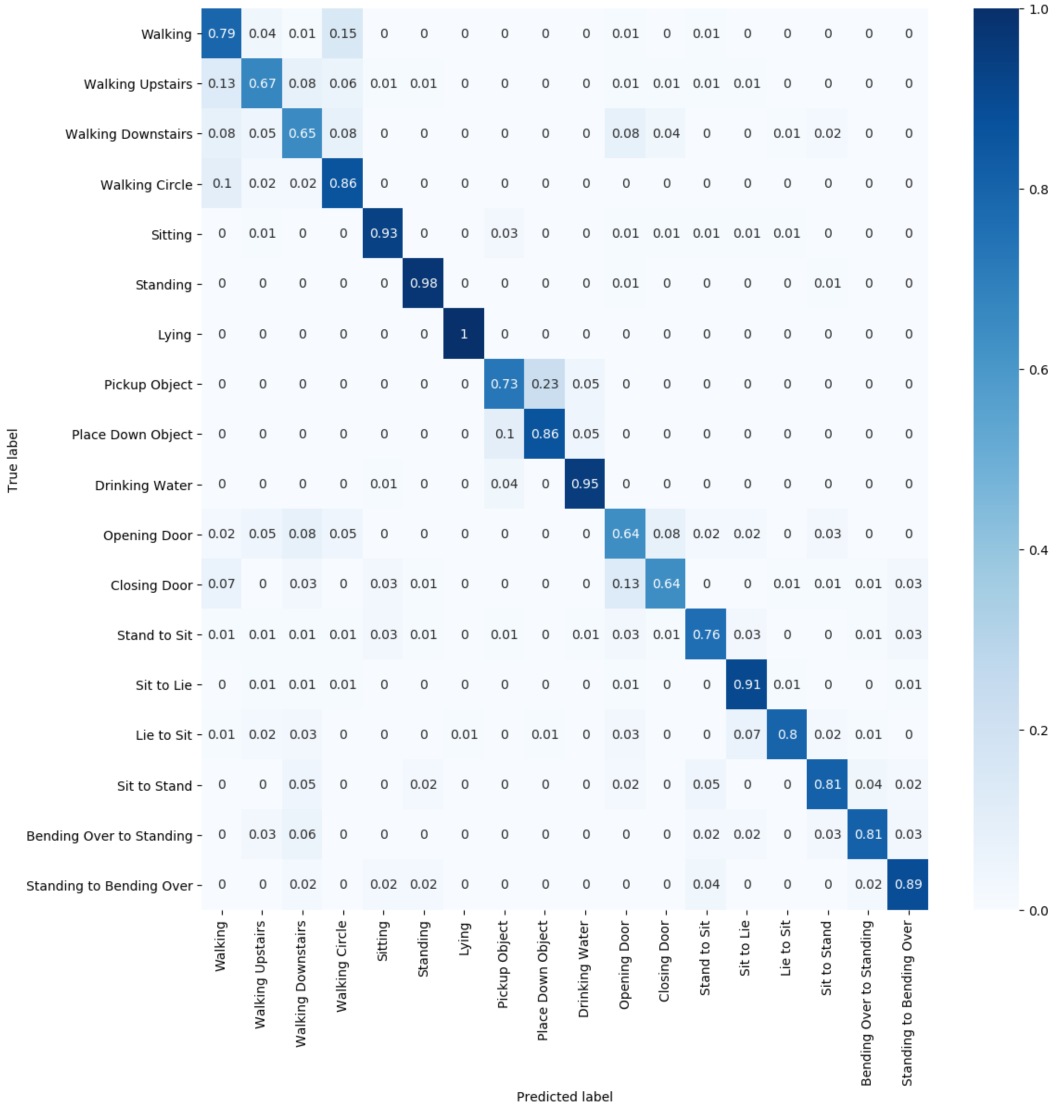


Wrist Confusion Matrix

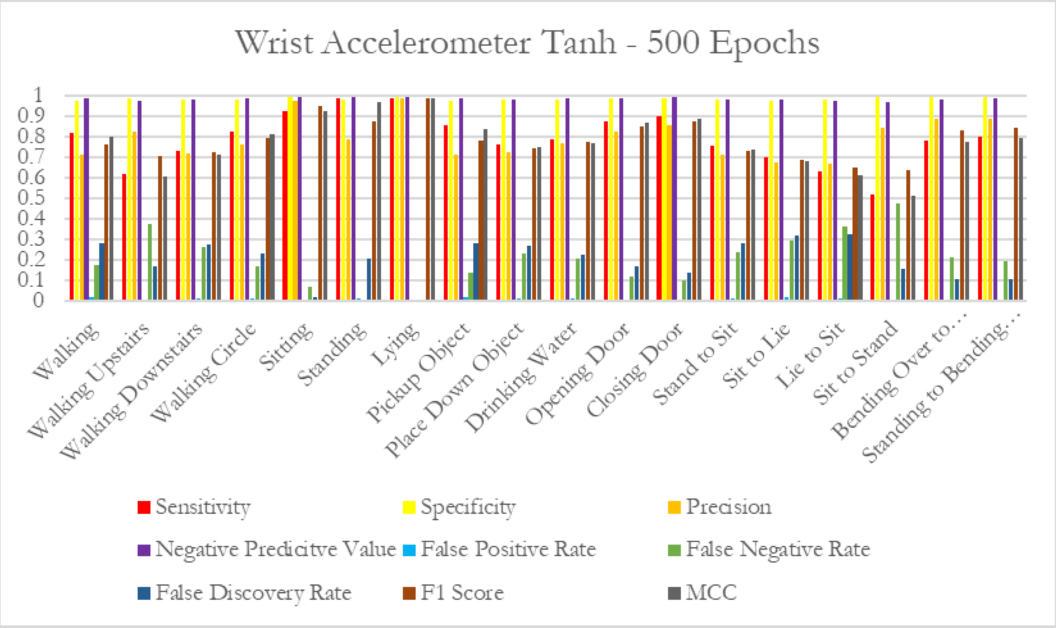


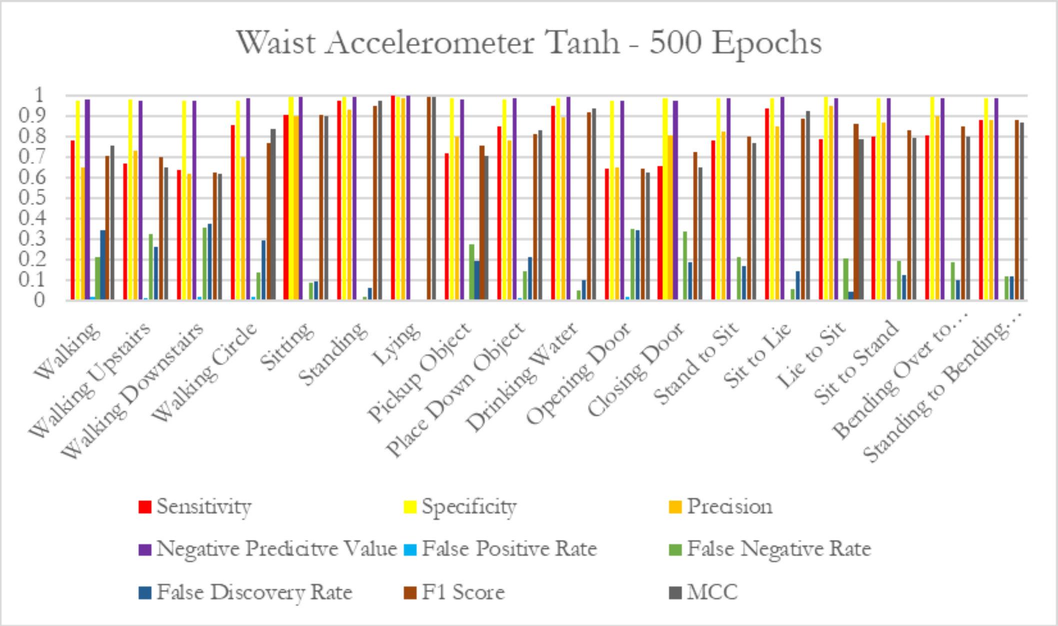
	Walking	Walking Upstairs	Walking Downstairs	Walking Circle	Sitting	Standing	Lying	Pickup Object	Place Down Object	Drinking Water	Opening Door	Closing Door	Stand to Sit	Sit to Lie	Lie to Sit	Sit to Stand	Bending Over to Standing	Standing to Bending Over
Walking	0.82	0	0.01	0.15	0	0.01	0	0	0	0	0	0	0	0	0.01	0	0	0
Walking Upstairs	0.04	0.62	0.07	0.01	0	0	0	0.01	0.02	0.02	0.03	0.02	0.03	0.05	0.06	0.01	0	0.01
Walking Downstairs	0.05	0.02	0.75	0.01	0	0	0	0.02	0	0	0	0.01	0.05	0.04	0.05	0	0	0.02
Walking Circle	0.12	0	0	0.83	0	0.01	0	0	0	0.01	0	0	0	0.02	0.01	0	0	0
Sitting	0	0.01	0.01	0	0.94	0	0	0.02	0.01	0	0	0.01	0	0	0.01	0	0	0
Standing	0.01	0	0	0	0	0.99	0	0	0	0	0	0	0	0	0	0	0	0
Lying	0	0	0	0	0	0	0.99	0	0	0	0	0	0	0.01	0	0	0	0
Pickup Object	0	0	0	0	0	0	0	0.86	0.07	0.05	0	0	0	0.02	0	0	0	0
Place Down Object	0	0	0	0	0	0	0	0.1	0.76	0.07	0.02	0	0	0.02	0	0.02	0	0
Drinking Water	0	0	0.01	0	0	0	0	0.09	0.1	0.78	0	0.01	0	0	0	0	0	0
Opening Door	0	0	0	0	0	0	0	0.02	0	0	0.88	0.07	0	0.03	0	0	0	0
Closing Door	0	0	0	0	0	0	0	0	0	0	0.1	0.9	0	0	0	0	0	0
Stand to Sit	0.03	0	0.01	0.01	0	0.04	0	0.01	0.01	0	0	0	0.75	0.03	0.03	0.03	0.01	0.03
Sit to Lie	0.01	0	0.06	0	0.01	0	0.01	0.03	0.03	0.01	0	0	0.04	0.69	0.09	0	0	0
Lie to Sit	0	0.04	0.03	0.03	0.01	0.01	0	0.02	0.02	0.03	0.03	0.03	0.03	0.09	0.64	0	0	0
Sit to Stand	0.04	0.04	0.04	0.04	0	0.04	0	0.02	0.02	0.04	0	0	0.11	0.02	0.05	0.54	0.02	0.02
Bending Over to Standing	0.03	0.02	0.05	0	0	0.06	0	0	0	0	0	0	0.02	0	0	0.02	0.79	0.02
Standing to Bending Over	0	0	0	0	0	0.09	0	0	0	0	0	0	0.02	0	0	0.02	0.07	0.81

Waist Confusion Matrix

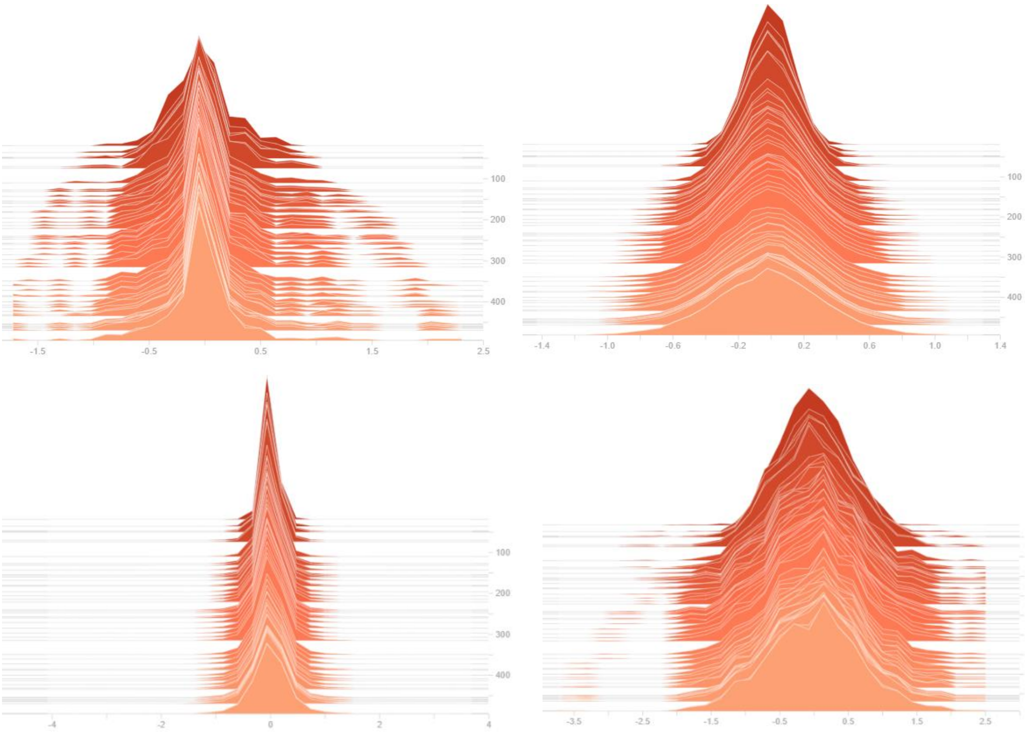


	Walking	Walking Upstairs	Walking Downstairs	Walking Circle	Sitting	Standing	Lying	Pickup Object	Place Down Object	Drinking Water	Opening Door	Closing Door	Stand to Sit	Sit to Lie	Lie to Sit	Sit to Stand	Bending Over to Standing	Standing to Bending Over
Walking	0.79	0.04	0.01	0.15	0	0	0	0	0	0	0.01	0	0.01	0	0	0	0	0
Walking Upstairs	0.13	0.67	0.08	0.06	0.01	0.01	0	0	0	0	0.01	0.01	0.01	0.01	0	0	0	0
Walking Downstairs	0.08	0.05	0.65	0.08	0	0	0	0	0	0	0.08	0.04	0	0	0.01	0.02	0	0
Walking Circle	0.1	0.02	0.02	0.86	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sitting	0	0.01	0	0	0.93	0	0	0.03	0	0	0.01	0.01	0.01	0.01	0.01	0	0	0
Standing	0	0	0	0	0	0.98	0	0	0	0	0.01	0	0	0	0	0.01	0	0
Lying	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Pickup Object	0	0	0	0	0	0	0	0.73	0.23	0.05	0	0	0	0	0	0	0	0
Place Down Object	0	0	0	0	0	0	0	0.1	0.86	0.05	0	0	0	0	0	0	0	0
Drinking Water	0	0	0	0	0.01	0	0	0.04	0	0.95	0	0	0	0	0	0	0	0
Opening Door	0.02	0.05	0.08	0.05	0	0	0	0	0	0	0.64	0.08	0.02	0.02	0	0.03	0	0
Closing Door	0.07	0	0.03	0	0.03	0.01	0	0	0	0	0.13	0.64	0	0	0.01	0.01	0.01	0.03
Stand to Sit	0.01	0.01	0.01	0.01	0.03	0.01	0	0.01	0	0.01	0.03	0.01	0.76	0.03	0	0	0.01	0.03
Sit to Lie	0	0.01	0.01	0.01	0	0	0	0	0	0	0.01	0	0	0.91	0.01	0	0	0.01
Lie to Sit	0.01	0.02	0.03	0	0	0	0.01	0	0.01	0	0.03	0	0	0.07	0.8	0.02	0.01	0
Sit to Stand	0	0	0.05	0	0	0.02	0	0	0	0	0.02	0	0.05	0	0	0.81	0.04	0.02
Bending Over to Standing	0	0.03	0.06	0	0	0	0	0	0	0	0	0	0.02	0.02	0	0.03	0.81	0.03
Standing to Bending Over	0	0	0.02	0	0.02	0.02	0	0	0	0	0	0	0.04	0	0	0	0.02	0.89

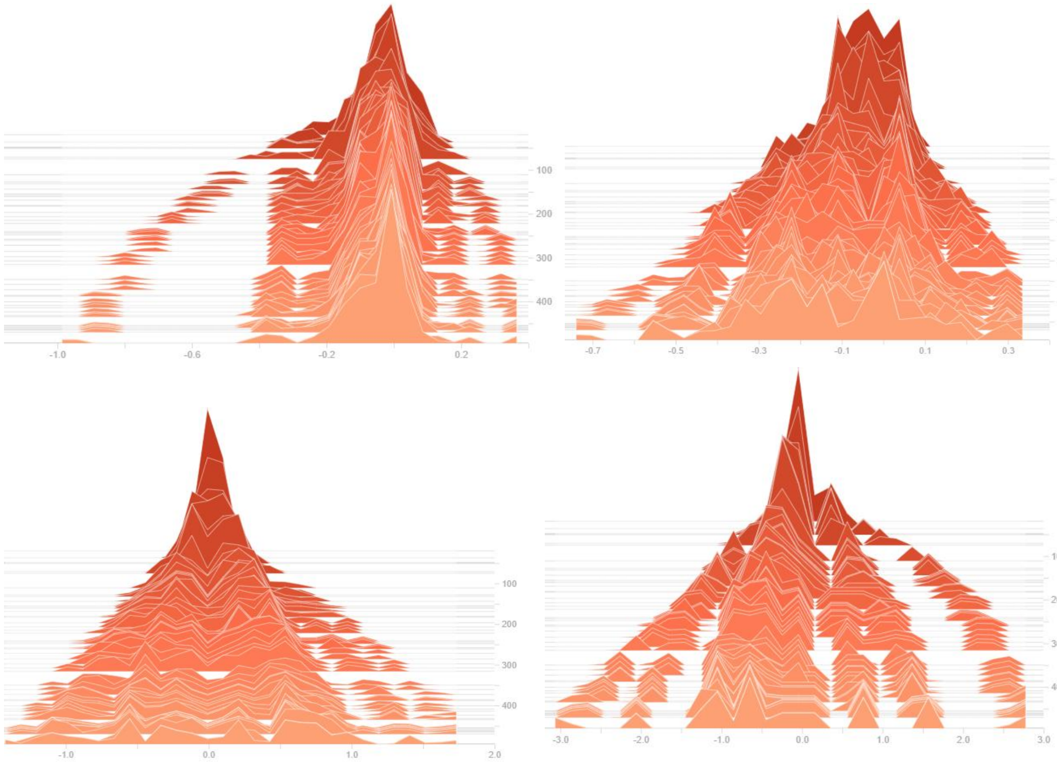




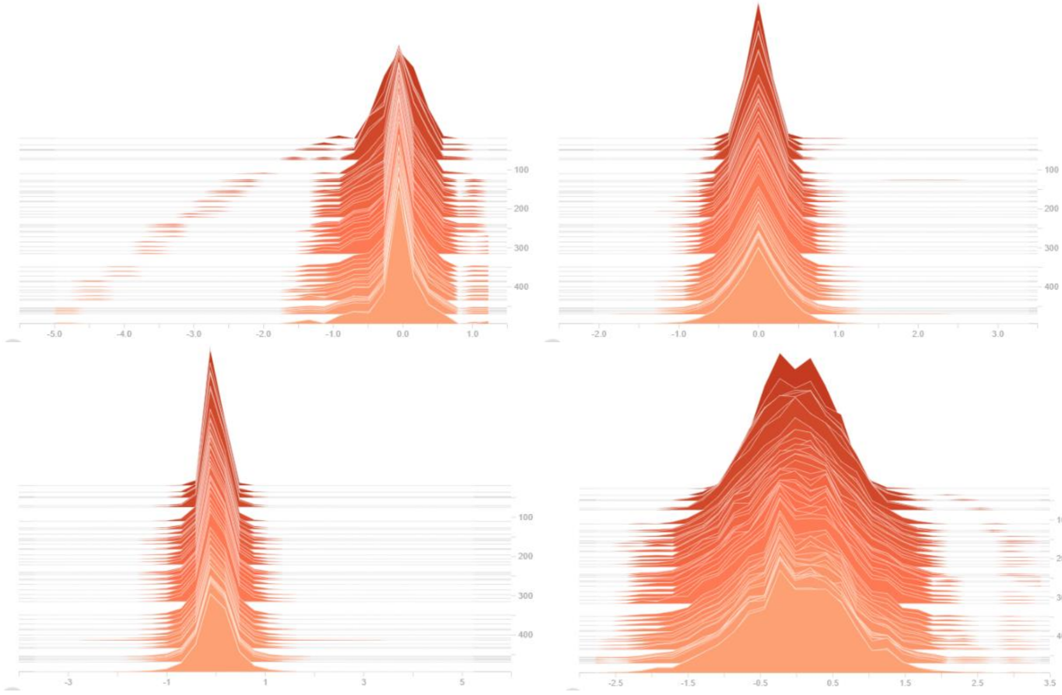
Wrist data set Histogram of kernel weights for the four layers of the network using Tanh at 500 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



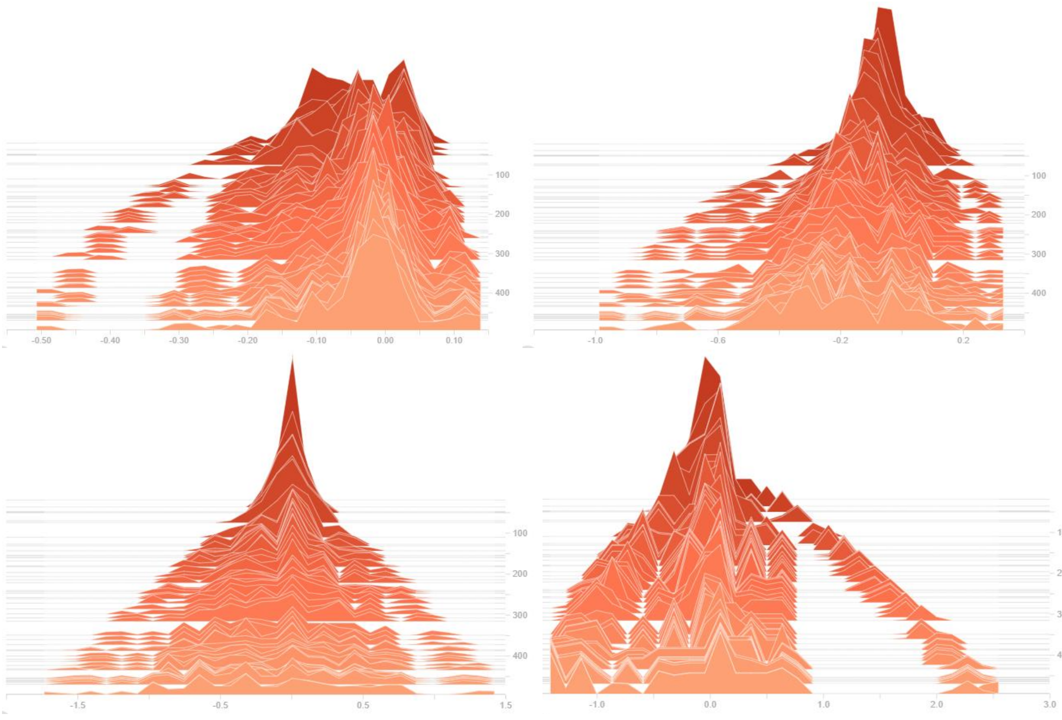
Wrist data set Histogram of bias weights for the four layers of the network using Tanh at 500 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



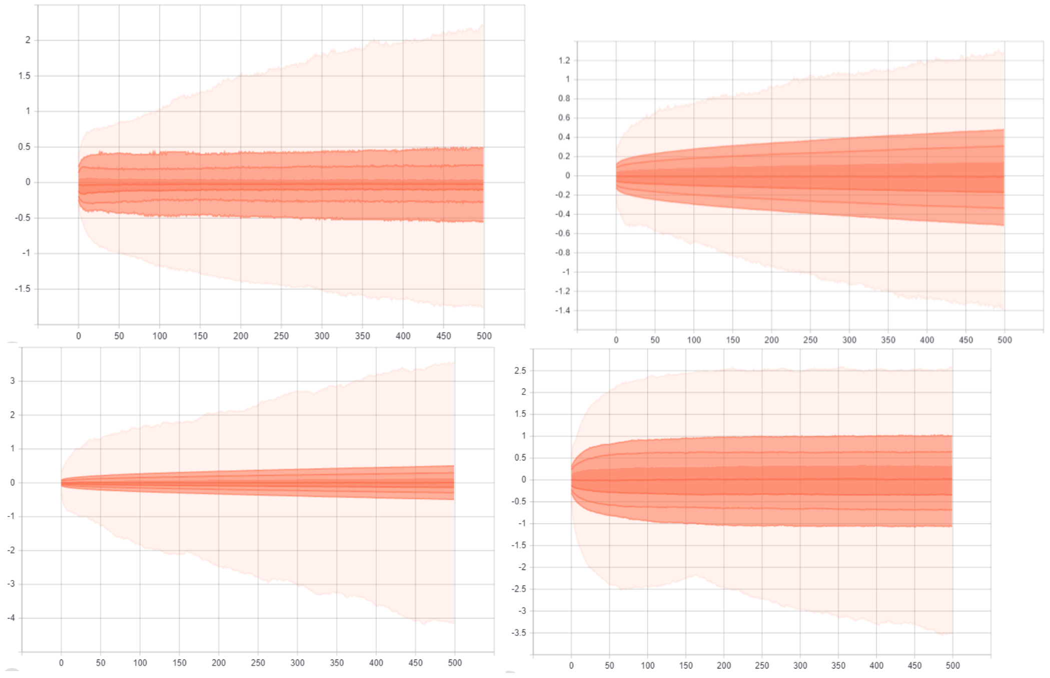
Waist data set Histogram of kernel weights for the four layers of the network using Tanh at 500 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



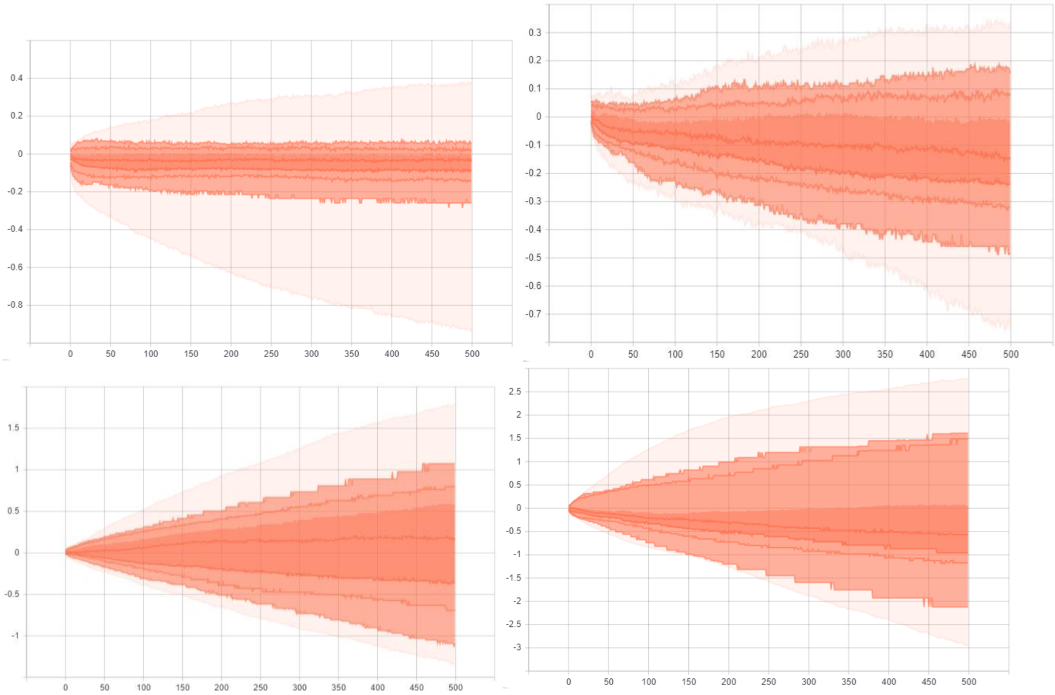
Waist data set Histogram of bias weights for the four layers of the network using Tanh at 500 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



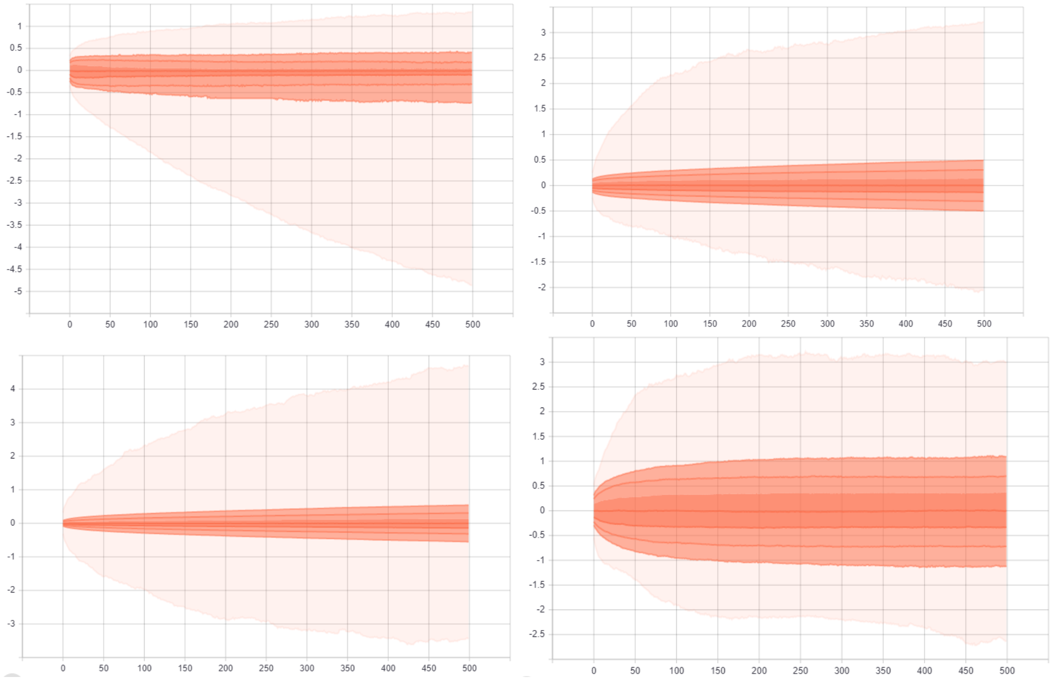
Wrist data set Distribution of kernel weights for the four layers of the network using Tanh at 500 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



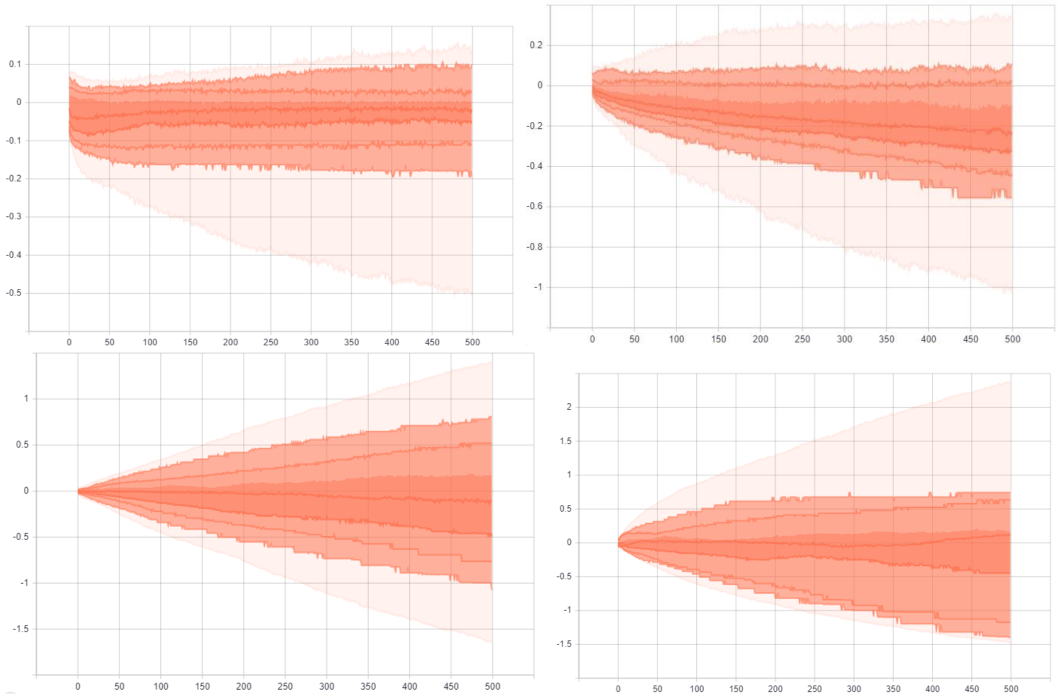
Wrist data set Distribution of bias weights for the four layers of the network using Tanh at 500 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



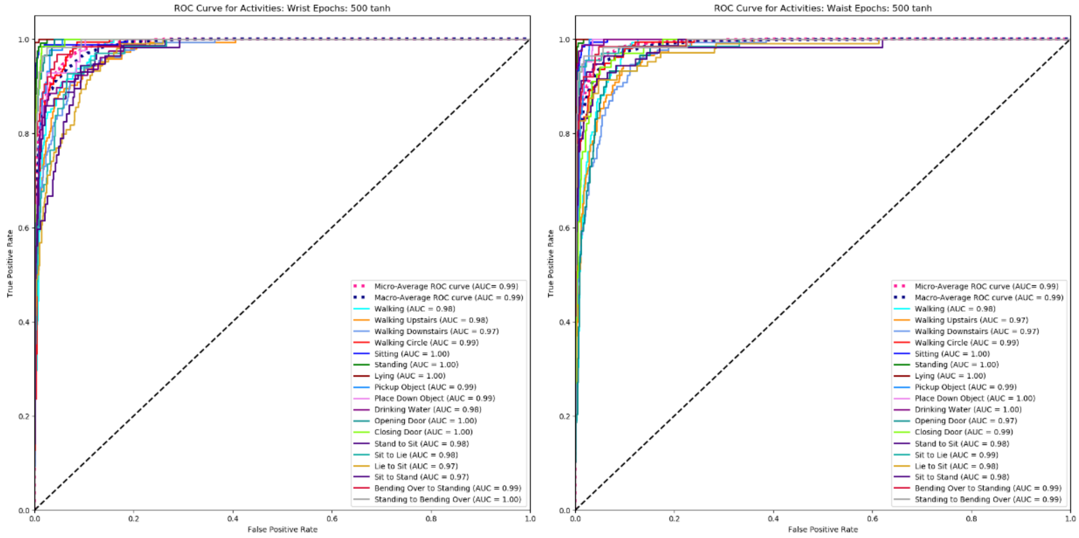
Waist data set Distribution of kernel weights for the four layers of the network using Tanh at 500 epochs. From top-left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



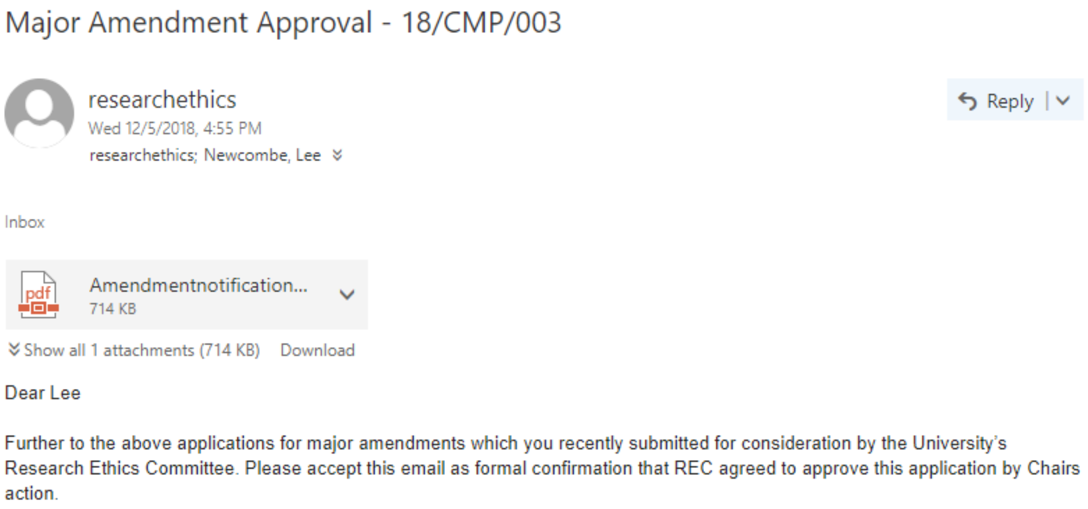
Waist data set Distribution of bias weights for the four layers of the network using Tanh at 500 epochs. From top-1left: Layer 1 Conv1D, Layer 2 Conv1D, Layer 3 Dense, and Layer 4 Dense.



ROC Curve for wrist and waist



Appendix Q – Ethical Approval Form



Major Amendment Approval - 18/CMP/003

researchethics
Wed 12/5/2018, 4:55 PM
researchethics; Newcombe, Lee

Amendmentnotification...
714 KB

Dear Lee

Further to the above applications for major amendments which you recently submitted for consideration by the University's Research Ethics Committee. Please accept this email as formal confirmation that REC agreed to approve this application by Chairs action.