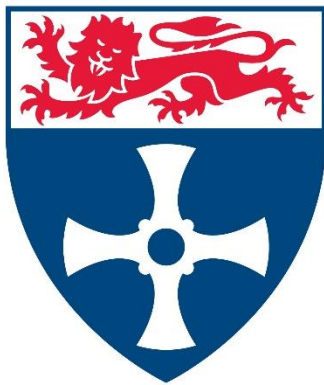


Newcastle University upon Tyne



Newcastle
University

Ravi s/o Suppiah

THESIS:

**Advancing Rehabilitative Robotics through Signal Processing
and Machine Learning Algorithms**

PhD Supervisors:

Dr Khalid Abidi

Dr Anurag Sharma

Dr Noori Kim

Table of Contents

Abstract	4
List of Figures	5
List of Tables	8
List of Publications	10
Chapter 1: Introduction	11
1.1 Robot-Assisted Therapy	12
1.2 Research Gaps and Limitations	12
1.3 Research Aim & Objective	16
Chapter 2: Literature Review	18
2.1 The Human Brain	18
2.2 Nervous System	20
2.3 Understanding Causes of Motor Movement Challenges	23
2.4 Robot Assisted Therapy	29
2.5 Neuroplasticity	30
2.6 Motor Learning and Neuroplasticity	32
2.7 Rehabilitative Robotics	36
2.8 Clinical Evidence	42
2.9 Control Algorithms	43
2.10 Review of Rehabilitative Robotic Devices	44
2.11 Sensors and Signal Processing	45
2.12 Classification and Machine Learning Models	61
2.13 Deep Learning	63
2.14 Edge Computing	65
2.15 Summary of Literature Review	68
2.16 Scope and Focus of this Research	69
Chapter 3: Novel EEG Model	70
3.1 The EEG Signal	70
3.2 Dataset	71
3.3 Methodology	72
3.4 Results and Discussion	86
Chapter 4: EMG Decoding and Classification	89
4.1 Physical Movements	89
4.2 Dataset	93
4.3 Signal Pre-Processing	95
4.4 Feature Extraction	97
4.5 Machine Learning & Classification	105
4.6 Unified Model	109
4.7 Results & Discussion	110
4.8 Conclusion	110

Chapter 5: FIS-LSTM Network	112
5.1 EEG Analysis & LSTM Architecture	112
5.2 LSTM Training & Testing	117
5.3 Fuzzy Logic	125
5.4 Applying Fuzzy Logic to EEG Signals	128
5.5 EMG Analysis	134
5.6 LSTM Training and Testing	137
5.7 Applying Fuzzy Logic to EMG Signals	143
5.8 Four-Way Classification using FIS-LSTM Model	149
Chapter 6: BIO-Inspired-Fuzzy-Inference-System (BIOFIS)	154
6.1 Motivation	154
6.2 Design & Implementation	156
6.3 Evaluation of BIOFIS on Physiological Data	161
6.4 Application of BIOFIS	164
6.5 Reduced Physiological Signals	165
6.6 Alternate Dataset	166
6.7 Discussion	169
6.8 Conclusion	170
Chapter 7: Edge Computing	171
7.1 Overview	171
7.2 Design & Implementation	172
7.3 Training & Finalizing the Network	173
7.4 MCU Deployment	175
7.5 Testing the Network	176
7.6 Improved Methodology	178
7.7 Discussion	185
Chapter 8: Discussion and Conclusion	187
References	190

Abstract

Rehabilitative robotics holds tremendous promise in improving the quality of life for individuals with motor impairments. The integration of signal processing and machine learning algorithms into rehabilitative robotics systems has emerged as a powerful approach to enhance the effectiveness and efficiency of rehabilitation therapies. This thesis aims to explore and contribute to the advancements in this exciting field.

The first part of this research focuses on signal processing techniques applied to the analysis and interpretation of sensor data in rehabilitative robotics. Various signal processing methods such as filtering, feature extraction, and time-frequency analysis are investigated to extract relevant information from sensory signals captured by robotic devices. These processed signals serve as valuable inputs for subsequent machine learning algorithms.

The second part of the thesis delves into the application of machine learning algorithms in rehabilitative robotics. Supervised, unsupervised, and reinforcement learning techniques are studied to model and predict user intent, adapt robot behaviour, and optimize rehabilitation exercises. These algorithms play a pivotal role in personalizing the rehabilitation process, enabling tailored interventions based on individual needs and progress.

The integration of signal processing and machine learning presents unique opportunities for real-time adaptation and closed-loop control in rehabilitative robotics. The combination of sensor data processing and machine learning enables the creation of intelligent robotic systems that can dynamically adjust therapy parameters, ensuring optimal engagement and challenging the user at an appropriate level. In addition to technological advancements, this research also addresses practical challenges in the implementation of signal processing and machine learning algorithms in real-world rehabilitative robotics applications. Considerations such as computational efficiency, robustness to noise and variability, and user acceptance are carefully examined to ensure the feasibility and effectiveness of the proposed approaches. Overall, this thesis aims to contribute to the field of rehabilitative robotics by advancing the integration of signal processing and machine learning algorithms.

List of Figures

Number	Title	Page
2.1	Regions and Directions of the Human Brain	19
2.2	Structure of a Neuron	21
2.3	Categories of Robot-Assisted Therapy	30
2.4	Sliding Theory using BCM Theory	32
2.5	Block Diagram for the use of EMG Signals	47
2.6	Three-Level Filter Bank	57
2.7	DWT Frequency Domain Representation	57
2.8	Multilayer Perceptron	61
2.9	SVM Hyperplane	62
3.1	Time and Frequency Domain Plots of C3 for the first 4s of Data Capture	73
3.2	Plot of C3 Channel Power Bands	74
3.3	Scatterplot of PSD Values	75
3.4	ERD/ERS Plot of C3	77
3.5	Scatter Plot and Confusion Matrix for kNN Classification using Alpha and Beta PSD features	78
3.6	Gamma Peak Observation around Rest/Active Transition	82
3.7	ERD/ERS with Gamma Peak Classification Model	84
3.8	Plot of C3 ERD/ERS Gamma PSD with Peak Detection Threshold of 1	85
4.1	Longitudinal and transverse representation of the forearm muscles.	92
4.2	EMG Sensor Placement	94
4.3	Sequence of ARM Movements	94
4.4	Raw EMG Signals	95
4.5	Raw, Filtered and Rectified EMG Data	96
4.6	Signal Pre-Processing Blocks	96
4.7	All EMG Channels, Raw and RMS	97
4.8	DWT Operation	98
4.9	Four-Level DWT Decomposition	98
4.10	Multi-Level Decomposition for AD Signal	99
4.11	Wavelet Decomposition of AD over 16 windows	100
4.12	ANN Model	105
4.13	Unified EMG Classification Model	109
5.1	LSTM Layer Architecture	113
5.2	LSTM Cell	114

5.3	CH3 Data for Trial Duration with Event Markers	116
5.4	CH3 Data for Forward and Reverse Movements	116
5.5	CH3 Data for GripUp and RelDown Movements	117
5.6	LSTM Performance for Training and Testing	118
5.7	Confusion Matrix for Training and Testing Phase	119
5.8	LSTM Performance for Training and Testing Phase	120
5.9	Confusion Matrix for Training and Testing Phase	121
5.10	Alpha PSD Distribution for 50ms Window	123
5.11	Alpha PSD Distribution for 50ms Window	124
5.12	Alpha and Beta Overlays for Forward and Reverse Action	124
5.13	Types of Membership Functions	126
5.14	AND and OR operators for MFs	127
5.15	Defuzzification using Centroid Method	128
5.16	MF of Alpha PSD	129
5.17	Complete FIS Model and Surface Plot	130
5.18	Alpha and Beta PSD Score	132
5.19	Results of FIS+LSTM Model	133
5.20	EMG Sensor Placement	134
5.21	Sequence of Actions performed by Participant	135
5.22	Raw EMG Data with Time Markers	136
5.23	RMS EMG Data with Time Markers	137
5.24	LSTM Performance for Training and Testing Phase	138
5.25	Confusion Matrix for Training and Testing Phase	139
5.26	LSTM Performance for Training and Testing Phase	141
5.27	Confusion Matrix for Training and Testing Phase	142
5.28	EMG RMS Values for all 5 sensors	143
5.29	MF of AD Sensor	144
5.30	Complete FIS Model and Surface Plot (Forward/Reverse)	145
5.31	AD, BR and FD channel RMS values and EMG Score (Forward/Reverse)	146
5.32	Complete FIS Model and Surface Plot (GripUp/RelDown)	147
5.33	FD, CED and FDI channel RMS values & EMG Score (GripUp/RelDown)	148
5.34	AUC Comparison of Deep Learning Methods	150
5.35	EMG Scores obtained through combined FIS model	150
5.36	Four-Way Classification Accuracies	151
5.37	Four-Way Classification Confusion Matrix	151
5.38	Four-Class Accuracy Comparison	152
6.1	BIOFIS Model	156
6.2	Generation of Fuzzy Rules based on input/output data	158
6.3	EEG Four-Class Accuracy for All Movements	163
6.4	EMG Four-Class Accuracy for All Movements	163
6.5	Feature Ranking of topmost EEG Features	164

6.6	Feature Ranking of EMG features for Forward/Reverse Movement	165
6.7	Classification Accuracy with Key Feature Reduced	165
7.1	Pin-Out Configuration on the STM Cube MX	175
7.2	Validation Analysis on the MCU Target	175
7.3	Confusion Matrix for two different pairs of movements	177
7.4	Comparison of Classification Accuracies between the PC-based system and the Embedded Controller	177
7.5	DWT Operation	180
7.6	Flow of Proposed Algorithm	181
7.7	RTOS Architecture	183
7.8	Scatter Plot for AD and BR Channel	184
7.9	Five-Way Confusion Matrix	184

List of Tables

Number	Title	Page
2.1	Comparison of Motor Learning Assessment	35
2.2	Comparison between End-Effector and Exoskeleton	37
2.3	Coarse Motor Rehabilitation	39
2.4	Rehabilitative Robotic Systems based on Arm and Finger Movements	41
2.5	Quantitative Summary of RT-based clinical evidence	42
2.6	Quantitative Summary of TD features	53
2.7	Quantitative Summary of FD Features	54
2.8	Motor-Imagery based BCI Systems	59
2.9	Applications of BCI-based Systems	61
2.10	Machine Learning Techniques and their Results	63
2.11	Comparison of Deep Learning Techniques	65
2.12	Comparison of Microcontrollers	67
2.13	Comparison of TinyML Framework	67
3.1	Correlation Coefficient Analysis for PSD	76
3.2	Plots and Values for various PSD Combinations	78
3.3	Plots and Values for various ERD/ERS Combinations	80
3.4	Results for Theta-Alpha-Gamma ERD/ERS features	82
3.5	Results for Alpha-Beta-Gamma ERD/ERS features	83
3.6	Results for ERD/ERS with Gamma Peak	86
3.7	Comparison of Results	87
4.1	Entropy values for entire duration	99
4.2	AD Windowed Entropy	103
4.3	BR Windowed Entropy	103
4.4	FD Windowed Entropy	103
4.5	CED Windowed Entropy	104
4.6	FDI Windowed Entropy	104
4.7	Input Data and Label for 'Rest' and 'Active' State	105
4.8	Classification Accuracy of Individual Channels	106
4.9	Classification Accuracy Summary	108
4.10	Classification Accuracy of Unified Model	110
4.11	Comparison of State-of-the-Art	111
5.1	Cell State Components of LSTM Network	113
5.2	Activation Formulae for Cell State Components	114
5.3	Time and Frequency Domain Features	122
5.4	LSTM Classification Accuracy (Forward & Reverse)	122
5.5	LSTM Classification Accuracy (GripUp & RelDown)	123

5.6	Rule Base for Alpha and Beta PSD Values	129
5.7	Forward/Reverse Classification Accuracy for RMS Data	140
5.8	GripUp/RelDown Classification Accuracy for RMS Data	142
5.9	Rule Base for AD, BR and FD RMS Values (Forward/Reverse)	144
5.10	Rule Base for AD, BR and FD RMS Values (GripUp/RelDown)	147
5.11	Classification Accuracy for Various Movements	150
5.12	Accuracy and F1-Score	152
6.1	Terms used in BIOFIS Model	157
6.2	Summary of Feature Selection and Classification Data for EEG	161
6.3	Summary of Feature Selection and Classification Data for EMG	162
6.4	Highest Classification Accuracies for Alternate Datasets	167
6.5	BIOFIS Ablation Study for EEG Features	168
6.6	BIOFIS Ablation Study for EMG Features	168
6.7	BIOFIS Ablation Study for EEG Features with C3 removed	169
6.8	BIOFIS Ablation Study for EMG Features with AD removed	169

List of Publications

An Electromyography-aided Robotics Hand for Rehabilitation – A Proof-of-Concept Study

2020 IEEE REGION 10 Conference (TENCON)

2020-11-16 | Conference paper

DOI: [10.1109/tencon50793.2020.9293940](https://doi.org/10.1109/tencon50793.2020.9293940)

A Novel Event-Related Desynchronization/Synchronization with Gamma Peak EEG model for Motor State Identification

2021 International Conference on Computational Science and Computational Intelligence (CSCI)

2021-12 | Conference paper

DOI: [10.1109/csci54926.2021.00245](https://doi.org/10.1109/csci54926.2021.00245)

Motor State Classification based on Electromyography (EMG) Signals using Wavelet Entropy and Neural Networks

2021 International Conference on Computational Science and Computational Intelligence (CSCI) 2021-12 | Conference paper

DOI: [10.1109/csci54926.2021.00248](https://doi.org/10.1109/csci54926.2021.00248)

Fuzzy Inference System (FIS) - Long Short-Term Memory (LSTM) Network for Electromyography (EMG) signal analysis

Biomedical Physics & Engineering Express | 2022-10-27 | Journal Article

DOI: [10.1088/2057-1976/ac9e04](https://doi.org/10.1088/2057-1976/ac9e04)

Part of ISSN: [2057-1976](https://doi.org/10.1088/2057-1976)

A comprehensive review of Motor Movement Challenges and Rehabilitative Robotics

Journal of Smart health | Vol 29 | 2023-09-23 | Journal Article

DOI: <https://doi.org/10.1016/j.smhl.2023.100402>

Bio-Inspired Fuzzy Inference System (BIOFIS) – for Physiological Signal Analysis

IET Cyber-Systems and Robotics | Vol 5 Issue 3 | 2023-07-10 | Journal Article

DOI: <https://doi.org/10.1049/csy2.12093>

Real-Time Edge Computing Design for Physiological Signal Analysis and Classification

Submitted: Biomedical Physics & Engineering Express

Under Review

Chapter 1

Introduction

Rehabilitative robotics, a multidisciplinary field that combines robotics, signal processing, and machine learning, has emerged as a powerful approach to improving the rehabilitation process for individuals with motor impairments. By integrating sophisticated signal processing techniques and machine learning algorithms into robotic systems, researchers have been able to enhance the effectiveness, efficiency, and personalization of rehabilitation therapies. This introduction provides an overview of the advancements in rehabilitative robotics achieved through the integration of signal processing and machine learning algorithms, and outlines the objectives and contributions of this thesis.

The Need for Rehabilitative Robotics

Stroke is the third leading contributor that causes disability in the long-term in many countries [1, 2]. It can have far-reaching effects such as speech impairment, paralysis, cognition impairment, memory loss, coma or even death. One of the most devastating effects of stroke is the impairment of mobility skills in the upper extremity (UE). This can have a significant impact on one's Quality of Life (QoL). Studies in neuroscience show that neuroplasticity is an effective means of helping stroke patients to retrain themselves to perform basic tasks that may have been impaired due to a stroke [3]. Through repetitive training, it is possible to create new pathways between neurons to restore the lost motor skills.

Task-oriented repetitive training has been shown to be an effective means of restoring the impaired motor skills [4]. The neuroplasticity phenomenon has been shown to be present in individual from birth through adulthood [5]. Over the years, much work has been done to take advantage of this understanding to make full use of neuroplasticity in the rehabilitation

process. In conventional therapy, patients have to undergo numerous hours of rehabilitation training with physiotherapists. This adds a great deal of cost and may also not be a practical solution for the care-givers and patients. Furthermore, the evaluation of the progress is very subjective and dependent on the therapist.

1.1 Robot-Assisted Therapy (RT)

Due to the shortcomings of the conventional therapy, the notion of Robot-Assisted therapy has taken on a greater spotlight, with many researchers looking into this area [6]. For a start, robot-assisted therapy provides significant reduction in cost and reliance on manpower [7]. It also holds distinct advantages in its ability to provide an objective and quantifiable measure of performance of the user [8]. This is due to the ability of such systems to measure several parameters, such as, reflex, level of voluntary control and range of movement. It enables much more accurate training evaluation which in effect helps to cater a more personal training programme for the patient [9-12].

1.2 Research Gaps and Challenges

Motor impairments resulting from various conditions such as stroke, spinal cord injuries, and neuromuscular disorders significantly impact an individual's ability to perform daily activities and reduce their overall quality of life. Traditional rehabilitation methods often rely on labour-intensive, therapist-led interventions, which can be time-consuming and costly. Rehabilitative robotics offers a promising alternative by providing automated, precise, and repetitive training with the potential for continuous monitoring and feedback. This technology has the potential to augment traditional therapy and extend rehabilitation opportunities beyond the clinical setting.

1.2.1 Limitations

One of the main advantages of robot-assisted rehabilitation systems is its ability to deliver training with high intensity and repeatability. However, there is no clear consensus on clinically accepted robotic structures for effective rehabilitation. The effectiveness of the design is very much based on large clinical trials [13].

It is also important to be able to quantify if the effects observed during the training sessions can be transformed into improvements in the motor skills of the patients and their ability to perform ADLs in the real world [13, 14].

There is also a need for adaptive control algorithms that can adjust and control the training movement repetitively and in a more natural way [15]. The effects of brain neuroplasticity are also not being fully taken advantage of, in the current systems [16].

1.2.2 Research Gaps

For Activities of Daily Living (ADL), the movements are generally dynamic and abnormal muscle synergies in stroke patients interfere with their movement goals and restrict their movement. The initial literature review reveals some gaps in this area that need to be addressed.

Multisensory Control and Feedback

It is important the motor control loop is closed, so that the patient is able to have a clear understanding of his progress. This can be achieved through visual and audio means, and also through EMG feedback. With the aim of enhancing Neuroplasticity, it is also important to incorporate EEG feedback through BCI systems. This will allow the patients to understand the importance of the brain in the rehabilitation process.

Whole Arm & Fine Motor Skills Training

Many robotic systems have demonstrated effectiveness in guiding major body parts such as shoulder and elbow. However, many ADL's require fine motor control, such as being able to turn a door knob, or use a spoon. These are still challenging tasks that require good control of the wrist, hand as well as independent fingers. Devices that are able to fulfil these requirements with a high-level of comfort and safety are in demand.

Assist-as-Needed Control

The main rationale behind this is to ensure that rehabilitation is provided only as much as needed to complete the activity without overdoing it. This is important as providing too much

assistance may result in negative consequences for the motor learning process. Some work has been done in the area of self-adaptive algorithms. However, they are still lacking an autonomous system that is able to adaptively configure the exercise parameters and settings to guide the user automatically.

Brain Neuro-Plasticity

The highly complex brain has the ability to adapt and learn based on the inputs that it receives. A learned action can also be relearned if the nature of the input signals changes. This is the reason why individuals, through various types of physiotherapy and repetitive actions, re-learn how to perform basic actions after experiencing some motor movement challenges due to an accident or illness. A robotic rehabilitative system should be able to mimic this neuroplasticity of the human brain for it to be effective and applicable for a wide range of users.

1.2.3. Challenges

Developing rehabilitative robotic systems poses several challenges that need to be addressed to ensure their effectiveness and successful implementation. Some of the key challenges include:

1. **User Variability:** Individuals undergoing rehabilitation exhibit a wide range of impairments, capabilities, and preferences. Designing robotic systems that can accommodate and adapt to this variability is a significant challenge.
2. **Safety and Robustness:** Ensuring the safety of users during robotic interactions is paramount. Robotic systems must be robust enough to handle unexpected events or user behaviour, minimizing the risk of injury or accidents.
3. **Real-time Adaptation:** Real-time adaptation is critical for tailoring rehabilitation interventions to each individual's changing needs and abilities. Developing adaptive algorithms and control strategies that can continuously monitor user performance and adjust therapy parameters accordingly is a challenge.

4. **Human-Robot Interaction:** Promoting natural and intuitive interaction between humans and robots is crucial for user engagement and motivation. Designing effective human-robot interfaces and interaction modalities that are user-friendly and easy to use can be challenging.
5. **Ethical Considerations:** Integrating robotic systems into rehabilitation raises ethical concerns related to privacy, autonomy, and the potential replacement of human therapists. Addressing these ethical considerations and ensuring the responsible use of technology in rehabilitative settings is essential.
6. **Cost and Accessibility:** Cost-effective and accessible robotic systems are necessary to ensure widespread adoption and availability of rehabilitative robotics. Developing affordable and easily deployable solutions that can be used in various settings, including hospitals, clinics, and home environments, is a challenge.
7. **Long-term Efficacy and Adaptation:** Assessing the long-term efficacy of rehabilitative robotic interventions and understanding how the therapy can adapt as the user progresses is a challenge. Longitudinal studies and personalized approaches are needed to optimize long-term outcomes.
8. **Integration with Existing Rehabilitation Practices:** Seamless integration of robotic systems with existing rehabilitation practices and workflows is crucial for their successful implementation. Collaboration with clinicians, therapists, and healthcare professionals is essential to ensure the compatibility and acceptance of rehabilitative robotic systems.
9. **Regulatory and Reimbursement Issues:** Compliance with regulatory standards and navigating reimbursement policies are challenges faced in the development and adoption of rehabilitative robotic systems. Addressing these issues is necessary for the widespread acceptance and sustainability of these technologies.
10. **User Acceptance and Engagement:** Ensuring user acceptance and engagement is critical for the success of rehabilitative robotic systems. Overcoming scepticism, building trust, and promoting the benefits of robotic-assisted rehabilitation are challenges that need to be addressed.

Addressing these challenges requires a multidisciplinary approach, involving collaboration between engineers, clinicians, therapists, and researchers. By addressing these challenges, rehabilitative robotic systems can be developed to effectively support and enhance the rehabilitation process, ultimately improving the outcomes and quality of life for individuals with motor impairments.

1.3 Research Aim & Objective

One of the primary advantages of integrating signal processing and machine learning algorithms into rehabilitative robotics is the ability to personalize and adapt rehabilitation therapies based on individual needs and progress. By continuously monitoring user performance through sensor data, the robotic system can dynamically adjust therapy parameters, such as assistance level, resistance, or task difficulty, to ensure optimal engagement and challenge. This personalized approach promotes active participation, motivation, and neuroplasticity, leading to more effective and efficient rehabilitation outcomes. To achieve such an outcome, this research aims to identify effective methods in encouraging a deeper level of involvement from the patients that involves feedback, assist-as-needed approach and adaptive control. This encompasses the research gaps that have been earlier identified and explained.

This research aims to achieve the following outcomes:

- Investigate the use of physiological signals such as EEG and EMG to accurately decode and classify a range of real-world actions.
- Investigate and evaluate various signal processing techniques for the analysis and interpretation of physiological data. This includes filtering methods, feature extraction algorithms, and time-frequency analysis tools.
- Exploring the application of different machine learning algorithms, such as supervised, unsupervised, and reinforcement learning, for modelling user intent and optimizing rehabilitation exercises.

- Develop a novel inference system that is able to mimic the neuroplasticity of the brain such that it is able to accurately decode and classify motor actions based on the available physiological signals.

In conclusion, rehabilitative robotics holds great promise for improving the quality of life for individuals with motor impairments. The integration of signal processing and machine learning algorithms into robotic systems offers the potential for more effective, efficient, and personalized rehabilitation therapies. This thesis aims to contribute to the field by investigating and advancing the use of signal processing and machine learning algorithms in rehabilitative robotics. Through this thesis, we aim to enhance and inspire research in the application of these novel techniques and algorithms in the field of rehabilitative robotics.

Chapter 2

Literature Review

2. 1 The Human Brain

The human body is a complex organism and the brain is its master. The brain weights only three pounds but carries many advanced operations. It is the main intelligence mechanism in the body which has to perform interpretation of senses, initiate movement of the body, and control the behaviour of the individual. It is protected with a soft bony shell and lies amongst a fluid that helps protect it. It is very much the pride and glory of the human body [17].

The brain has always been a fascination amongst philosophers and scientists. For the longest time, they viewed as something too complex to comprehend. However, advances in Science and Technology in the past 30 years have enabled us to get a peek into the internals of the brain. Great progress in the areas of behavioural science and neurological studies have helped to accelerate our understanding of this crown jewel. In the USA, the 1990s was also terms to be the “Decade of the Brain”. Many research organizations all over the world see the potential of the brain and have set-up their own research centres to focus on this research. The cumulative effort of all these research groups has helped us achieve a far deeper understanding of the human brain than was ever thought possible.

2.1 1. Brain Anatomy

The brain weighs just 3 pounds, but yet it essentially controls all our bodily functions by continuously gathering information from the external world, and processing it to provide the appropriate response. The brain is protected by the skull and comprises the brainstem, cerebellum and cerebrum. The cerebellum is connected to the spinal cord through the brain stem, and so is the cerebrum. These function as a relay centre.

Our five senses of hearing, taste, touch, sight and smell generate signals for the brain. This happens frequently with multiple such signals being sent to the brain simultaneously. The data is packaged into messages in a way that allows the brain to interpret it into a meaningful output. These messages can also store information into our memory. Many aspects of our life like our thoughts, speech, memory, organ functions and motor control movements are controlled by the brain.

2.1.2. Brain Regions

Figure 2.1 shows the main regions and directions of the human brain [18]. The four visible lobes are Frontal, Parietal, Temporal and Occipital. The two hidden lobes are Insula and Medial Temporal Lobe.

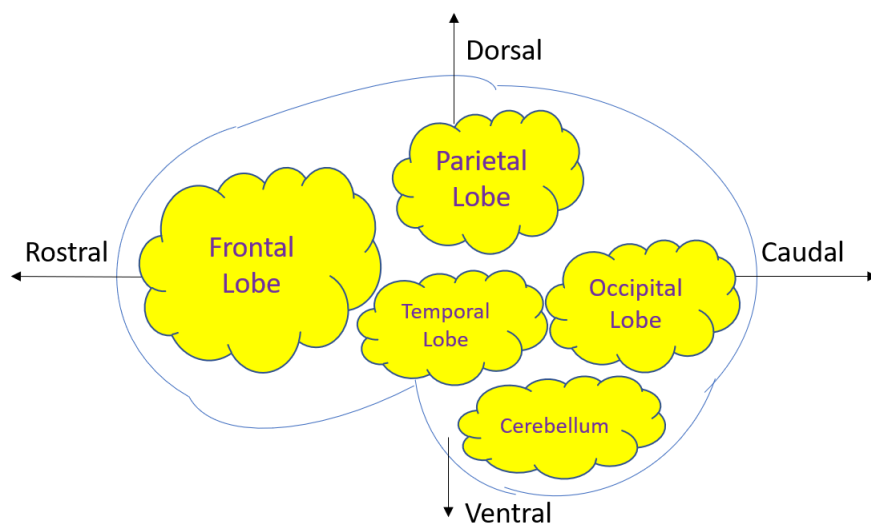


Figure 2.1: Regions and Directions of the Human Brain

The activities of higher-level cognition, motor skills, expressive language and reasoning are associated with the frontal lobe. Near the vicinity of central sulcus, at the rear of the frontal lobe, lies the motor cortex. Information is sent to the brain through various lobes of the brain so that it can process it and carry out the appropriate motor movements. When there is any form of damage or injury to the frontal lobe, many effects can be observed. These include changes to the attention capacity, sexual habits, interpersonal communication and behaviour, as well as being more prone to risk.

The brain's middle section is where the parietal lobe is found. Tactile sensory data, such as those associated with pain, touch and pressure are processed in this area. Within this particular lobe, there is a separate portion labelled the somatosensory cortex. The senses of the body are processed here. If there is any form of damage or injury to the parietal lobe, there can be challenges in the areas of language, eye gaze, as well as verbal memory.

In the bottom area of the brain, the temporal lobe is positioned. Interpretation of the sounds and language are processed in the primary auditory cortex which is located within the temporal lobe. The ability to form memories is linked with the hippocampus and this is also located within the temporal lobe. If there is any injury or damage to this section of the brain, it can potentially lead to challenges in language, perception of speech and memory.

Interpretation of the data obtained through visual input and its associated processing is carried out by the occipital lobe. This is located towards the back of the brain. Data obtained from the eyes' retinas are channelled to the primary visual cortex which is found within the occipital lobe. Any injury or damage to the primary lobe will cause effects in the identification of colours, recognition of words and identifying objects.

2.2 Nervous System

The nervous system is made up of many critical organs which include the spinal cord, the brain, sensory organs and the extensive array of nerves that inter-connect all the various organs of the body. The central nervous system (CNS) is formed by the spinal cord and the brain. This is where the vast array of information from the various organs is processed and a decision is made. The peripheral nervous system (PNS) is made up of sensory nerves and sense organs that monitor the state of the various organs and send this information to the CNS.

Within the nervous system, there are cells of two classes. They are neuroglia and neurons.

Neurons (Figure 2.2) communicate within the body by transmitting signals through the electrochemical form. Neurons have a unique look due to the long cellular processes that play a significant role in fulfilling their function. Most of the other cellular organelles, together with the mitochondria and nucleus, are contained within the cell body. Dendrites are small structures that resemble a tree-type structure. They extend away from the cell body into the

surrounding environment to pick up stimuli from sensory receptor cells or other neurons. Effector Cells and other neurons pick up signals transmitted through the Axons of neighbouring neurons.

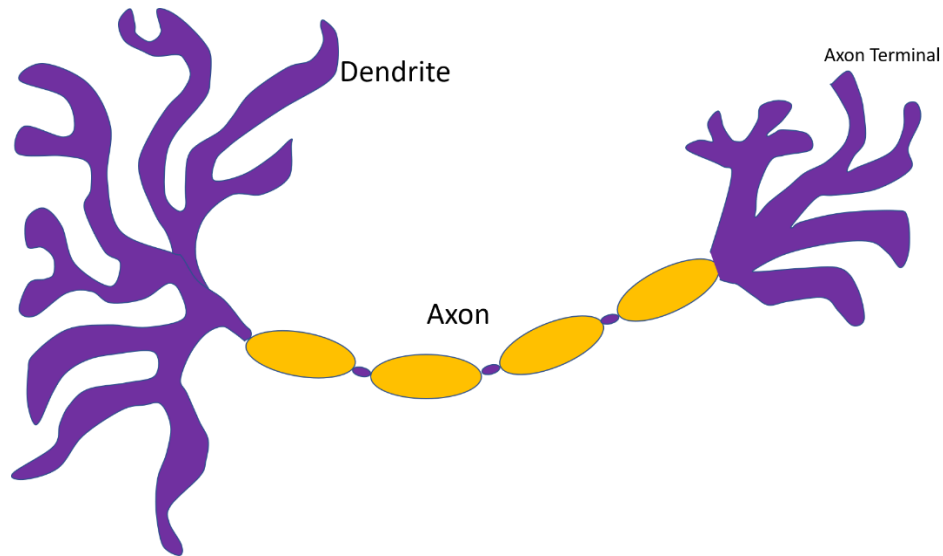


Figure 2.2: Structure of a Neuron [19]

The nervous system has sub-divisions, namely the Somatic Nervous System (SNS), Central Nervous System (CNS), Autonomic Nervous System (ANS), Peripheral Nervous System (PNS), , and the Enteric Nervous System (ENS).

The brain and the spinal cord together form the Central Nervous System (CNS). It acts as the central command centre of the entire body by providing critical functions related to memory, processing and regulation. There is an extensive array of subconscious and conscious sensory data that is generated throughout the body. The CNS acts as the central point where all this information is relayed. Based on the data that it receives, the CNS makes decisions about the actions it must take, maintain the homeostasis of the body, and in general, ensure its survival. The CNS also takes responsibility for higher level brain functions like emotions, expressions, creativity, language and personality. It very much determines and shapes our unique individuality.

The Peripheral Nervous System (PNS) comprises of all the other areas within the nervous system, excluding the brain and the spinal cord.

The Somatic Nervous System (SNS) is a portion of the PNS that comprises the efferent neurons that are voluntary. Voluntary conscious effort effectively control the SNS, and is generally responsible for the stimulation of the skeletal muscles that are in the body.

The Autonomic Nervous System (ANS) is a part of the PNS that comprises the efferent neurons that are involuntary. It controls subconscious activities that involve effector neurons, such as gland tissue, muscle tissue and cardia tissue.

The Enteric Nervous System (ENS) is a part of the ANS that takes charge of the controlling the functional aspects of the digestive organs as well as regulating digestion.

2.2.1 Action Potential

Neurons transmit information (signals) throughout the body through a process known as action potential. This is a part of the process that takes place whenever a neuron is fired. Concentration gradients play a key role in supporting and triggering this phenomenon in neurons. It refers to the difference in the concentration of the ions between the inside and outside of the neuron. As long as there is a higher concentration of a particular charged ion on one side of the neuron as compared to the other, a large concentration gradient would be formed. Regardless of the charge of the ions, both polarity ions will move in directions that would eventually balance the gradient.

2.2.2 Resting Membrane Potential

Most of the time, neurons have a negative concentration gradient. This is possible, as outside the cell, there are positively charged ions in high concentration. This regular state is also referred to as a resting membrane potential. When the neuron is in this state, the number of sodium ions Na^+ is higher outside as compared to inside. Similarly, there is also an increased number of potassium ions K^+ inside as compared to outside. The concentration of ions isn't constant and the neurons constantly flow in and out of the neurons to equalize and maintain the negative concentration gradient.

2.2.3 During Action Potential

When there is a sudden movement of ions in and out of the neuron, a resultant change in the neuron's membrane potential is realized, which is effectively the action potential. When a cell body sends out an impulse, the sodium channels will open and there is an increase in the inflow of sodium ions into the cell. At some point, a threshold will be reached, which will cause an action potential to trigger (firing of neurons), and this sends the signal down its axon. It is based on an all-or-nothing law, where the neurons fire or they don't. There is no in-between. This behaviour allows nerves to fire at their full-strength for the signal to be propagated to neighbouring neurons.

2.2.4 After Action Potential

Once the neuron has completed firing, it goes through a refraction cycle. During that period, it is not possible to have one more action potential. After this cycle, the neuron will be able to return to its normal resting potential. It is then possible to have another action potential.

It is the firing of neurons in the brains that we aim to capture and assist us in decoding the signals to get a better understanding of how they reflect our thoughts and actions.

With this understanding of the human brain, we will now proceed to look at Stroke and its effects.

2.3 Understanding Causes for Motor Movement Challenges

The two broad reasons for muscular issues are stroke and muscular system diseases. We will discuss both of them to better understand their causes and effects.

2.3.1 Stroke

The brain is filled with arteries, and they are affected in any way, it can lead to a stroke. In the USA, it is one of the main causes of disability. It is also the fifth highest cause of death. The blood vessels in the brain supply it with nutrients and oxygen. If it bursts or some clot is

formed, a stroke occurs. When a portion of the brain is unable to get sufficient oxygen and blood, the brain cells will die [20]. For this reason, Stroke is also referred to as a cerebrovascular disease.

2.3.1.1 Causes of Stroke

- **Ischemic Stroke (Clots)**

When a blood vessel supplying blood to the brain faces some form of obstruction, it causes an Ischemic stroke. About 87 percent [20] of the reported strokes belong to this category.

Atherosclerosis is the condition when fatty deposits start to line the walls of the arteries. This is the main cause for such type of strokes. The deposit of fatty lining can cause 2 types of obstruction:

Within the vessel of the blood, the fatty plaque can cause a blood clot and this is known as Cerebral Thrombosis.

In other cases, a blood clot can form in some other location within the body. This is known as cerebral embolism. This is generally the heart, as well as large arteries within the vicinity of the neck and upper chest. A portion of the blood clot dislodges and enters the circulatory system which allows it to travel to the blood vessels in the brain. At some point, the vessels in the brain may be too small to allow it to go through. One of the main causes of this is a heartbeat that is irregular. This is also known as embolism.

- **Silent Stroke**

A silent stroke is also known as a silent cerebral infarction (SCI). In many cases, the person experiencing this may not know that it is actually a stroke. It is most likely related to a blood clot which interrupts the flow of blood in the brain.

The irregular beating of the heart, also known as atrial fibrillation, enhances the chance for a person to get SCI. There are also other risk factors such as hypertension, high systolic blood pressure and increased levels of blood homocysteine. Early detection and treatment can help prevent the onset of a silent stroke.

- **Hemorrhagic Stroke**

Approximately 13% of strokes are caused by this haemorrhagic stroke. When a blood vessel weakens, it has the possibility to rupture and cause bleeding in the surrounding brain area. As the blood start to accumulate in the brain area, it causes compression in that part of the brain.

Intracerebral Haemorrhage and subarachnoid Haemorrhage are the dual types of haemorrhagic strokes. The main cause of such strokes are weakened blood vessels which can be classified as arteriovenous malformations (AVMs) and aneurysms.

In some cases, blood vessels that are abnormally formed can create a cluster and this is known as an arteriovenous malformation (AVM). When one of the vessels ruptures, the bleedings within the brain will begin. For aneurysm, a part of the blood vessel that is weakened will start to balloon. If it is not detected and treated, it can also eventually rupture and cause bleeding in the brain.

- **TIA (Transient Ischemic Attack)**

Another name for a mini-stroke is Transient Ischemic Attack (TIA). However, it should be taken as a major warning. The nature of this is temporary, where the blood flow to the brain may be affected for a short period of time. Since there is no permanent damage, many people tend to ignore it. However, this is very dangerous as a TIA may be precursor to a full-fledged stroke later on.

Some of the warning signs that should not be ignored are:

- Dizziness
- Constant Pounding headache without any particular reason
- Blindness in one or both of the eyes
- Slurred speech or challenges in understanding and analysing
- Weakness or numbness in part of the body.

- **Cryptogenic Stroke or Stroke of "Unknown Cause"**

A blood clot that obstructs blood from reaching the brain is the most common cause of a stroke. However, there are always instances where even after extensive testing, the actual cause for the stroke can't be determined. Such strokes are classified as being "cryptogenic".

About 25% of stroke survivors tend to have a similar stroke event in the future [20]. By finding the root cause of the stroke, appropriate treatment and preventive measures can help to lower the likelihood of experiencing another one. When the cause of stroke is not known, it can be quite stressful for the patient. However, with proper follow-up with the medical team, it is possible to make critical changes in your lifestyle to help prevent another stroke from occurring.

- **Brain Stem Stroke**

When a person has brain stem stroke, it is a challenge to diagnose and the patient tends to experience complex symptoms which include dizziness, vertigo and high levels of imbalance. These usually occur together without the typical symptom of experiencing weakness on a particular side of the body. It can also cause slurred speech, blurred vision and a lack of consciousness.

2.3.1.2 Effects of Stroke

There are multiple effects of stroke that can range from impairment of speech, memory loss, paralysis, loss of ability to reason, and in more severe cases, coma and even death [21]. One of the impairments that is commonly reported is that of the loss of partial or complete ability to move the Upper Extremity (UE). This generally causes a distinct negative impact on a person's quality of life as it affects the ability of the person to perform the Activities of Daily Living (ADL) [22]. Stroke and its impact is very subjective as it depends greatly on the individual, the severity, type of stroke and any other medical conditions the person may already have. Rehabilitation is critical to minimize the chance of these effects being permanent.

Repetitive training that is task-oriented, can help to enhance and potentially restore the motor skills that had been impaired by a stroke [23]. This is possible due to the neuroplasticity of the brain. That is its ability to change and adapt through re-learning [24]. Neural pathways can be restored and motor skills can be improved through targeted rehabilitation techniques. This can help improve the ADL for patients with Spinal Cord Injuries (SCI) and stroke. In recent decades, there has been a better understanding of the brain and this has helped in the development of tools and devices that can aid in repeated, targeted movements of the body. Together with that, there has also been much attention in the area of robot-assisted rehabilitation.

2.3.2 Muscular System Disease

Such diseases affect the ability of the user to control the muscles and can also cause issues with the nerves. In some cases, the disease is a side-effect of another condition.

2.3.2.1 Muscular System

Muscle's play a very important role in our Activities-for-Daily-Living (ADL). These include a wide range of activities like standing, sitting, walking, eating, talking, and even blinking. The muscles in our body, more than 600 of them, can be broadly classified into three types.

- **Skeletal Muscle**

These are tissues that consist of tough cords and are generally connected to the bones. They are also supported through tendons that play a significant role when the muscles contract and expand. They are also linked to the bone structure through ligaments and play an important role in maintaining the form of the skeletal structure of the body.

- **Smooth Muscle**

The internal organs of the body, such as the intestines, stomach and blood vessels are made up of smooth muscles. The actions carried out by these muscles are involuntary but are very critical to fulfil the needs of the function.

- Cardiac Muscle

This is found in the heart and its main role is ensuring that the contractions within the heart are coordinated to allow it to pump blood to all parts of the body efficiently. It consists of muscles that are made out of stretched fibres.

2.3.2.2 Muscular System Diseases

Such diseases can generally be classified as Primary or Secondary.

Primary Muscle Diseases

- **Polymyositis (PM)**

Also known as inflammatory myopathy, it causes inflammation of the muscle and the surrounding tissue, as well as blood vessels. Patients may experience challenges in recovering from falls and even have difficulty swallowing. It is treatable, even though there is no known cure.

- **Dermatomyositis (DM)**

It causes rashes, weakness of the muscles and pain. There is no known cure, but it can be treated with therapies and medications.

- **Muscular Dystrophy (MD)**

It causes loss of muscle mass and weakness. It is an inherited disease and its onset can occur at any age. Once the disease starts to be active, it becomes worse over time and eventually, the patients lose the ability to walk. Treatments like therapy and orthopaedic devices can help to improve their ADL.

Secondary Muscle Diseases

These diseases are generally a side-effect and appear alongside another health condition. These can include things like metabolic disorders, vascular diseases, infectious diseases and endocrine diseases.

Causes of Muscular Diseases

One of the main causes is genetics, where it can be inherited or due to a mutated gene. In some cases, it can also be classified as an autoimmune disease where an individual's immune system starts attacking its own muscle tissues. Other causes include injury, cancers, infections, medications, etc.

Diagnosis

The Electromyography (EMG) is the most pervasive tool that can help a medical professional assess muscular diseases. This can be accompanied with specific tests administered by physiotherapists to gauge the level of the issue.

Treatment

Medications and Therapy are well established methods to treat such patients. Therapy done with supervision and guidance will be most helpful in ensuring the patient is performing the correct actions so as to achieve the desired outcome.

2. 4 Robot-Assisted Therapy

Traditional therapy methods have several shortcomings, such as its high cost, the need for extensive labour, and the challenges of evaluating the performance and effectiveness of such therapies. Robot-Assisted Therapy (RT) is a field that has been growing in recent years to overcome these shortcomings and provide a more holistic rehabilitation experience to patients [7]. Neuronal plasticity is based on the ability of the brain to establish pathways to close the sensory motor loop. Multisensory feedback is critical to achieve this objective [25]. There has been a wide range of multi-sensory feedback techniques that have been investigated. These include Brain Computer Interface (BCI) technology [26], Virtual Reality (VR) [27], Electromyography (EMG) feedback [28], and Haptic Stimuli [29], amongst many others. The neuron rehabilitation process is achieved through the multisensory feedback which is enabled through the use of robotics and appropriate electrical stimulation.

In this research, a detailed literature review was executed to understand the state-of-the-art that is currently in this field, clearly identify the research gaps and look at the potential areas

for future work. Figure 2.3 shows the main categories of the papers that were selected for this research. The selection was made based on the alignment with the current research focus.

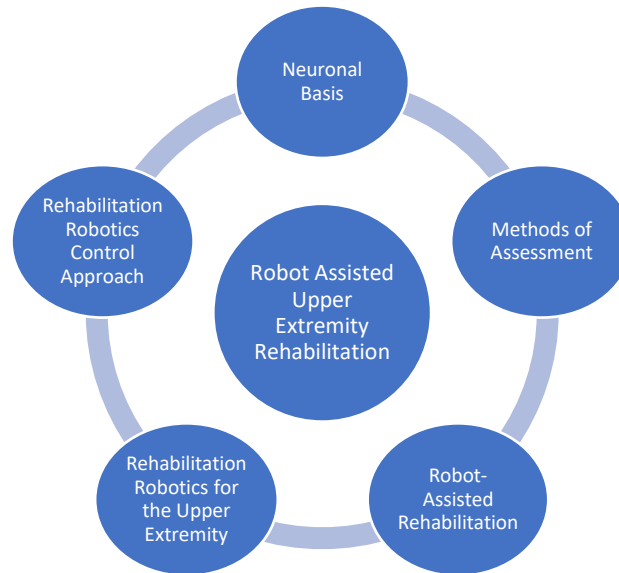


Figure 2.3: Categories of Robot-Assisted Therapy

2.5 Neuroplasticity

Animals have complex brain interconnections that are able to absorb vast amounts of information, process it and take meaningful action according to their needs. The interconnections between the various sections of the brain are fully formed at birth. They have to be learned over time, and this learning journey varies according to the species and the tasks involved.

Neuroplasticity is the ability of the brain to adapt itself by varying the connection strengths between its neurons [30]. Neural Networks present a flexible and powerful framework for solving many complex challenges. Their ability to mimic the brain's neuronal structure by developing a set of weights with varying strengths is its key idea. Finding the correct weights and its internal connections has been proven to be challenging.

In the area of supervised learning of a feed-forward network, both the output and input of a feed-forward network are earlier known, and error propagation is used to train the network

[31]. Though this approach has been largely successful, it is not something that is plausible in the biological sense as it requires backwards propagation [32-35].

For unsupervised neural learning, we do not specify the network's output. Utilizing the statistical characteristics of the input data, the network is able to tune itself and present the output. Many methods have been shown to train networks using this approach [36, 37]. Alternatives to this were also proposed in [38].

2.5.1 Hebb Learning

In [39], the manner in which the synaptic strengths between neurons changed was hypothesized. The Hebb Theory supported the notion of long-term synaptic potentiation (LTP) in which there is a consistent increase in the synaptic strength between neurons as long as they were continuously stimulated [40]. However, the Hebbian formula demonstrated two clear limitations [41]:

- The synaptic weights are uni-directionally positive. They can only increase monotonically.
- The potentially large values for the synaptic weights did not make much sense for biological applications.

2.5.2 BCM Theory of Metaplasticity

The most well-known and widely accepted alternate theory to the Hebbian formula was the BCM Theory [42]. Similar to Hebb, repeated firing of neurons strengthens their synaptic weights. However, the weights will be continuously regulated in an associative and dissociative manner. It offered a new approach which made use of a sliding threshold mechanism to maintain stable and realistic levels of the weights. Figure 2.4 illustrates this concept.

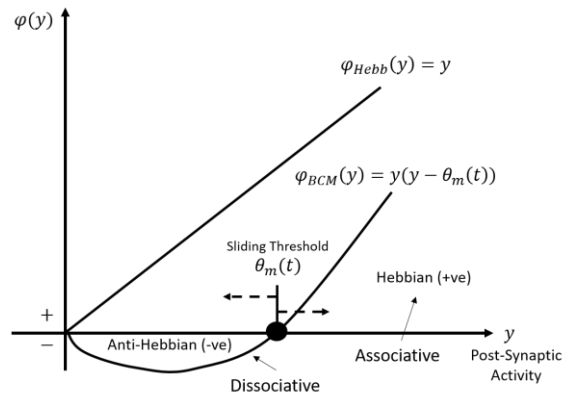


Fig 2.4: Sliding Theory using BCM Theory [42]

2.6 Motor Learning and Neuroplasticity

Evidence from various brain analysis techniques have indicated that neuron organization occurs as a patient undergoes rehabilitation after stroke, and this is regardless of the age of the patient [43]. These include functional magnetic resonance imaging (fMRI), transcranial magnetic stimulation (TMS) and the transcranial direct current stimulation (tDCS). Repetitive training is a key-enabler to aid in motor skills restoration of patients [44].

Traditionally, neurological-focused rehabilitation therapies were designed to be bottom-up. At the lower-level (bottom), physical movement was exerted with the aim of influencing the neuronal system (top). In recent years, the approach has been reversed. In [45], tDCS and TMS was used to aid in the rehabilitation of a hand that was impaired. In [46], Functional Electrical Stimulation (FES) was evaluated to analyse its effectiveness in improving a patients grasping and reaching functions. The results of this research showed that mere repetitive movements alone are not sufficient and motivation was a key factor to aid in the progression of such patients [47]. People are generally motivated when there is positive feedback and they are able to see some tangible effects of their work [48]. Where training is task-specific and goal-oriented, the results are more lasting [49].

2.6.1 Motor-Learning Assessment

There are a wide range of quantitative evaluation methods to assess the progress of motor skills through the rehabilitation process.

Action Research Arm Test (ARAT)

The upper extremity is assessed using this methodology. An ordinal score on 19 items is used. Where there is no movement, a score of 0 is assigned. Normal movement is assigned a score of 3. A summation is then carried out to create four subscale scores which consist of the following: gross motor (maximum of 9pts), grasp (maximum of 18 pts), grip (max of 12 pts), and pinch (max of 18pts). The maximum score an individual can get will be 57, and this indicates normal performance [50].

Fugl-Meyer Assessment (FMA)

The Fugl-Meyer Assessment is a performance-based, stroke-specific index for impairment. A wide range of assessment is performed in areas like balance, joint range of motion, sensation, motor functioning, and joint pain. The scale consists of 5 domains with a total of 155 items. Scoring is based on direct observation of the patient's performance. A 3pt ordinal scale is used where 2 points is given when the task can be fully performed, 1 point when it is partially performed and 0 points when the task cannot be performed [51].

Motor Status Scale (MSS)

The Motor Status Score measures wrist, finger and hand movements (with a maximum score of 42) and the measurement of the shoulder and elbow (maximum score of 40) [52]. It provides an expansion of the upper extremity measurement defined by FMA.

Wolf Motor Function Test (WFMT)

The WFMT is a method that is based on time and is used to perform an evaluation of the performance of the upper extremity. It also provides insights into the total limb and joint-specific ability [53].

Blocks and Box Test (BBT)

The Blocks and Box Test requires an individual to move blocks between different compartments in a box. The assessment is based upon the time taken to perform this task. The validity and reliability of this methodology has been well established [54].

Jebsen-Taylor Hand Function Test (JHFT)

The JHFT provides an objective and standardized evaluation of gross and fine motor hand movements based upon Activities of Daily Living (ADL). The assessment is done based only on speed and not the performance quality [55].

Nine-Hole Peg Test (NHPT)

The NHPT was designed with the aim of measuring finger dexterity. It is a relatively inexpensive test and can be carried out quickly. It should be used together with other tests to get a clearer picture of the functional ability of the patient [56].

Table 2.1: Comparison of Motor Learning Assessment

Method	Estimated MCID	Strengths	Weaknesses	Time
MSS [52]	Upper Extremity 66 points	It assesses the whole UE which includes the fingers, hand, wrist, forearm, elbow and shoulder.	Lack of precision in evaluating fine hand movement.	15-20 min
UE-FMA [51]	Finger, wrist and hand (max score = 42). Shoulder and Elbow (max score = 40).	Valid and reliable assessment of UE disability and impairment.	Challenging and consumes a lot of time.	20-25 min
WMFT [53]	Chronic Stroke: 1.5 – 2s Acute Stroke: 19s	Standard instructions. Applicable at various stages of the recovery	Takes longer time	30 min
BBT [54]	6 blocks using the affected hand.	Easy and quick.	Requires a minimal distal control	5-10 min
ARAT [50]	Chronic Stroke: 6 pts Acute Stroke (dominant hand): 12 pts Acute Stroke (non-dominant hand): 17 pts	Fast and Easy to test. Applicable at various stages of the recovery.	Non-commercial product. Need to self-build.	10 -15 min
NHPT [56]	Using affected hand: 38s	Quick and Cheap	Better for individuals who are on a better recovery path	10 min
JHFT [55]	-	Standard Steps	Requires a minimal distal control	15-40 min

*MCID: Minimal Clinically Important Difference

2.7 Rehabilitative Robotics

Regardless of the level of impairment, a patient may use robotic-based systems for their therapy. Such systems allow us to obtain a quantifiable measurement of the performance of a particular task. It also helps in the savings and cost of therapists by easily enabling repetitive movements automatically [57, 58]. Traditional modes of therapy require numerous sessions which could make them unaffordable for many patients. At a reasonable cost, robot-assisted therapy can provide a practical solution to these patients [59]. They have the ability to be consistent with the accuracy, precision and measurement of the patient's tasks and can also assist in providing tactile feedback that can help to correct the patients' movements. Importantly, they can help in data collection and analysis that will help therapists have a clear view of their progress [27].

Robotic rehabilitation also opens up another area of development where it can be fused with other technical levers like Brain-Computer Interface (BCI), Virtual Reality (VR) or others [26, 29, 60]. Another key advantage is that robotics makes rehabilitation more accessible. It can be carried anytime, in any location, much to the convenience of the patient.

Though there are several advantages of robot-assisted therapy, there are still some challenges. Patients may not clearly understand the deliverables of a robotic-based system and have unrealistic expectations. This may de-motivate them in continuing with such a therapy. There must also be a means to effectively measure the extent to which such activities are carried out to achieve the optimum gain. Customization of such devices may also be a challenge as most commercial products may have only a limited scope for it. Patients need to be able to have a long-term view of their therapy and have a better understanding on how the robot-assisted therapy helps them in improving the activities for daily living (ADL). Safety is also a critical issue on their mind. Proper fail-safe mechanisms need to be built into the system at both the Hardware and Software level to ensure that any error or failure in the system does not have an adverse impact on the patient. In many cases, the high cost of such robotic systems can also be a deterrent for hospitals and clinics to invest in such methods of rehabilitation [61, 62]. These challenges facing the research and engineering community inspire innovation and creativity in their solutions. Only a small number of patients who

require these are actually able to get it, but the cost of those left behind create a significant impact to the rest of the community [63].

2.7.1 Rehabilitation Robotics targeted towards hand recovery

There are generally two main categories when designing such devices. They are either the exoskeleton or the end-effector. The end-effector device is focused on higher-level movements targeted at related ADL. There is usually a distal support that aids in the training and the proximal joints are not directly affected by the system. These could result in some abnormal or unnatural joint movements if the system is not carefully designed. These devices also tend to be more flexible as they cater to coarse-level movements. The exoskeleton is a wearable that is focused on closely following the joint layout of the hand. It helps to follow-through on the user’s movements, by providing some level of assistance. They tend to be more customized to closely match the dimensions of the user. As they are able to closely follow the joints movements, they are able to minimize any erroneous actions by the user. Due to the need to support more joint movements, exoskeletons tend to be more expensive compared to end-effectors. A comparison of these devices is summarized in Table 2.2.

Table 2.2: Comparison between End-Effector and Exoskeleton

	End-Effector	Exoskeleton
Movements	Coarse-Level	Fine-Level
Joint-Control	Not at fine motor level	Yes
Adaptability	Yes, easy to adjust and cater to individual	No, needs to be customized to each individual.
Cost	Low	High

These devices can also be categorized into active and passive. When the system doesn’t detect any voluntary muscular contraction, the robot may still be able to control the actions of the hand. These can include activities like reaching, grasping and holding. In the active mode, the patient is able to exert some level of muscular force, and, the system is able to

detect that intention to provide a certain level of assistance to the user [64]. The system can be designed to provide Assist-As-Needed (AAN) support to ensure that such actions also trigger the neuro-rehabilitation process.

2.7.2 Coarse Motion Rehabilitation Robots

Depending on the target size and area of the body, it is possible to classify rehabilitation as being coarse motion or fine motion. We will first look at some of the highly-cited work in this area.

A breakthrough system that was designed with the objective of integrating neurological rehabilitation with the physical aspect, was, MIT_MANUS [65]. Though it was a stable and safe system, its range of motion was limited to two dimensions, being focused on the elbow and shoulder regions. An improvement to that was seen in MIME [66], where there was an added control of the forearm. The ARM Guide [67] provided a system that was able to guide a patient to be able to move their arm on a path that resemble a straight line. Those earlier systems in the past, have now been overshadowed by more complex systems that are highly portable, light-weight and with more Degrees of Freedom (DOF).

The ARMin-II [68] is a 6-DOF exoskeleton that targets a wide range of body parts, like the wrist, forearm, elbow and shoulder. It is equipped with a variety of sensors to measure torque, force and position. It also provides audio-visual feedback to aid with the patient's therapy. HapticMaster [69] is a 3-DOF multi-sensory platform designed to be generic towards applications where there is a requirement for human interaction. It has high joint stiffness together with increased force sensitivity. NeReBot [70] is known to be a safe robot with 3-DOF which can be easily transported. Two industrial robots were integrated to create the REHAROB therapeutic system [71] with 12 DOF. Though it is able to provide a full range-of-motion of the elbow and shoulder, it is limited to passive training only. The table below summarizes the review of these systems. The CADEN-7 system [72] is an active-assisted robotics system that is able to provide a total of 7 DOF. It is equipped with built-in safety features and is able to avoid any erratic motion by the user. A summary of this work is available in Table 2.3.

Table 2.3: Coarse Motion Rehabilitation

Name	DOF	Inputs	Actuators	Target	Features
ARM [67]	3	Forearm position and torque	Magnetic Particle Brake and DC motor	Elbow and Shoulder	End-effector
InMotion [73]	ARM 3	Angular velocity and torque, joint positions.	Brushless motors	DC Elbow and shoulder	Active, Passive and Interactive
MIT-Manus [65]	2	Torque and Joint	Brushless Motors	DC Elbow and Shoulder	End-effector
Bi-Manu-Track [73]	1	Not mentioned	Not mentioned	Forearm and Wrist	Commercial system.
ARMin [68]	6	Torque, Force and Position	Not mentioned	Wrist, Forearm, Elbow and Shoulder	Haptic System with audio-visual display.
GENTLE/s [74]	6	Torque, Velocity and Position	Brushed motors	DC Elbow, Shoulder and Forearm.	Based on Haptic Master
HapticMASTER [69]	3	Torque, Velocity and position	Brushed Motors	DC Shoulder, Elbow and Wrist	Virtual Exercise with feedback
REHAROB [71]	12	End-Point Torque	Electrical Motor	Shoulder and Elbow	CORDIS Project
MIME [66]	6	Forearm	Brushed Servos	DC Shoulder, Elbow and Forearm	Can program trajectories
NeReBot [70]	3	Motor Positions	DC Motors	Shoulder, Elbow and Forearm	Wire Support
CADEN-7 [72]	7	Torque, Force, Position	Cable Motors	Pulley, Should, Elbow and Forearm	Active, Passive
L-EXOS [75]	5	End-Point Torque	Brushed Motors	Elbow, Shoulder, Forearm	End Effector

The Light Exoskeleton (L-EXOS) [75, 76] system focuses on allowing the patient to fulfil a particular trajectory with the aid of virtual reality. It is equipped with a wide variety of sensors and is able to provide 5 degrees of freedom.

2.7.3 Fine Motion Rehabilitation Robotics

Coarse motion rehabilitation systems are not able to handle issues related to the arm and hand stiffness. ADL's requires the ability to have better control of the fingers which is also not addressed by such systems. Systems designed to address these concerns are reviewed here.

Amadeo [4, 77] was verified to show improvements in pinching and grasping through the FMA arm motor scale and Jebsen-Taylor Test. Cybergrasp [78] is a glove with force-feedback designed for use in therapy sessions. The system was evaluated in virtual environments that presented a wide range of feedback. Another device aimed at helping patients by fusing various sensors and promoting motor relearning was the Hand of Hope. The InMotion HAND robot [79] is highly customizable system that can provide consistent training and feedback to the user. The Gloreha rehabilitation glove is lightweight and comfortable. It is able to support a wide range of exercises with a different motion range for the various fingers.

The Rutgers Hand Master [80] is a glove with force-feedback that is used together with a virtual environment. In HEXORR, we saw a system that was able to provide the full range-of-motion for all the fingers[12]. A cable-based device was observed in HandCARE, where the main focus was to provide assistance to the patient to help open and close the hand, while being reasonably flexible to accommodate a broad range of finger shapes and sizes [81].

There are also many commercial systems in the area of rehabilitative robotics. One of the pioneers is the ReoGo system, Though it was designed to assess and train the upper extremity of stroke patients [82], it was also able to provide a fair amount of training for the hand and finger movements. For patients who have lost the ability to voluntarily activate their arm muscles, the ArmeoPower [83] proved to a valuable device. RUPERT is a lightweight 5-DOF whole arm exoskeleton. It is fitted with pneumatic muscle actuators and is able to assist the patient with a wide range of motion like shoulder, elbow, forearm, and wrist. It is able to provide qualitative feedback to the user to aid in the therapy [84].

With four degrees of freedom, the Biomemetic Orthosis for Neurorehabilitation (BONES) [85] is inspired by the biomechanics of humans. It is an active rehabilitative device that is able to provide a variety of motion. A summary of these devices can be seen in Table 2.4.

Table 2.4: Rehabilitative Robotic Systems focused on Arm and Finger Movements

Name	DOF	Inputs	Actuators	Target Area	Features
Hand Mentor	1	Wrist angle	1 Pneumatic	Wrist, fingers	Wearable
HWARD	3	Joint angles	3 Pneumatic	Wrist, fingers	Stationary
Haptic Knob	2	Wrist, forearm	2 DC brushed motors	Forearm, wrist	Stationery
T-WREX	5	Join angles	None	Shoulder, fingers	Wheelchair mounted
ReoGo	2	Position of end-point	4 electric motors	Shoulder, elbow, forearm, fingers	End-effector based
Armeospring	7	Joint angles	Not specified	Whole arm	Commercial
ArmeoPower	6	Joint angles	6 DC Motors	Whole arm	Commercial
BONES	4	Joint angles	5 DC Motors	Upper Limb	Safety, End-Effector

It can be observed that there are many commercial and research-based devices in the area of rehabilitative robotics that can aid a patient's recovery. However, each system has its challenges which can be a hindrance to the general acceptance of such devices. A rehabilitative robotic system must be able to answer a few fundamental questions:

1. How can the patient achieve a natural movement, considering the trajectory and the force?
2. How can the robot provide the right level of assistance that is required by the patient?
3. What kind and frequency of feedback is necessary to enhance the therapy?
4. What are the safety and portability concerns?

2.8 Clinical Evidence

In Table 2.5, we review the literature which has demonstrated that RT-based therapy has been effective in a patient’s therapy and recovery process.

Table 2.5: Quantitative summary of RT-based clinical evidence

Device	Size	Baseline	Duration	Outcome	Conclusion
MIME	27	UE-FMA = 24.8/66 FIM = 54.5/63	2 months	FMA-proximal = $\Delta 3.3$ FIM = $\Delta 2.2$	Improvements seen
MIT-MANUS	56	FMA-WH = 0.0 MSS-SE=3.6	5 weeks	FMA-WH=1.0 MSS-SE=13.1	Improved Performance
REHAROB	12	FIM=106.875	20 days	FIM = $\Delta 6.7$	Decreased spasticity
ARM	19	FS=0.7	8 weeks	FS=0.79	Improvement
NeReBot	35	FMA-SE=8.0 FMA-WH=0.0	5 weeks	FMA-SE=12.8 FMA-WH=3.0	RT + CT is effective
T-Wrex	23	UE-FMA=24.0	8 weeks	UE-FMA = $\Delta 3.7$	Improvement
InMotion	62	UE-FMA=20.3	6 weeks	UE-FMA = $\Delta 2.94$	Improvement
GENTLE/s	31	UE-FMA=30.0	9 sessions	UE-FMA=45.0	Improvement

Abbreviations: UE: Upper Extremity, FMA: Fugl-Meyer Assessment, SE: Shoulder and Elbow Score, FIM: Functional Independence Measure, CT: Conventional Therapy, RT: Robot-Assisted Therapy.

Symbol: Δ increase in gain

2.9 Control Algorithms

From the human perspective, achieving complex multi-joint movements may seem like a trivial task. In essence, it involves the integration of multi-sensory information from various parts of the body, together with the processing of this information in the brain, to achieve such an outcome. Depending on the objective of the rehabilitation system, various control strategies can be implemented to suit the requirements.

- Passive Control

The patient has limited control of the movements and the robotic system dominates the actions. Full control is given to the system to allow the patient to achieve a desired objective.

- Active Control

The patient has a significant level of control on the desired action. The robot will aid the patient to move towards the final goal. It has been reported [47] that such active control strategies help in the reorganization of the neural-connectivity in the brain. This in-turn has a cyclic positive effect on the overall effectiveness of the physiotherapy.

Assist as Needed (AAN)

In active assistance/control, a key objective is always to provide the right amount of assistance as required to complete the task [86]. In these cases, the robot can assist and also help to correct the movement of the patient. To achieve this objective, the system must be able capture multi-sensory data and perform the necessary adjustments to the system while it is being used. To perform this task effectively, the system must employ algorithms that can detect the strain a patient is feeling while attempting an activity and attempt to provide the correct amount of assistance through the various actuators.

2.10 Review of Rehabilitative Robotic Devices

In this chapter, we have seen that there are wide variety of rehabilitative robotic devices, both in the research and commercial space. They have been designed to evaluate or assess the feasibility of such systems and the results achieved so far are promising. One of the key challenges still facing users is the ability to customize such devices to suit individual physical attributes, as most of these exoskeletons are made with rigid material that has limited scope for any fine-adjustment. Furthermore, there is also a lack of simple and effective Graphical-User Interface (GUI) for the user to carry out the required exercises with proper feedback and guidance. Another key element, is the lack of such systems to be able to log the patient's data and provide meaningful data analytics into the performance of the patient at the end of each therapy session. Such information will be highly beneficial to the patient and the physiotherapist.

There have also been fair number of reports [22, 57, 87, 88] that there was a lack of significant difference between conventional therapy and robotic therapy, and that in some of these cases the conventional approach showed better results. It was highlighted that there could be other factors such as the intensity, stage of recovery and duration, that can affect the efficacy of such robot-assisted therapy.

Besides that, a wide number of papers [6, 89-92] have observed improvement in the motor movement ability of the hand in the areas of grip strength, push and pull. These were mainly proven through the Fugl-Meyer Assessment (FMA) and the Functional Independence Measure (FIM).

With continued research and development in this area, we can very soon expect the proliferation of such devices in individual patients' homes.

2.11 Sensors and Signal Processing

Movement of any part of the body is representative of neuromuscular activities. Capturing such signals is critical to designing systems that act upon the intent of the user. They also play a key role in the rehabilitation assessment process and can help to provide a more objective assessment of a patient's recovery.

The human skeletal muscular framework was first modelled by in 1938 and was known as the Hill-type model [93]. Three main electrical elements of series, parallel and contractile were used to model and replicate the nature of human muscles. This model was used as the basis for the design and implementation of a exoskeleton arm in [94], and was used in subsequent research activities. However, the complexity of the several arm parameters made it challenging as they were very dependent on an individual's physiological characteristics.

When using EMG signals for rehabilitation, we are in understanding the neuromuscular activation that is observed in the various muscles during different activities, such as, functional movements, activities of daily living, and training. There are two main classes of resistance-training exercises, and they are isometric and isotonic contractions. In the field of rehabilitation, both types of concentrations are widely used, while isotonic contractions are commonly observed in areas where the focus is on athletic and strength goals.

2.11.1 Isotonic

In isotonic contractions, we will observe muscular contractions that oppose resistance, and the length of the muscle will also change. By varying the muscle's length, the contraction is able to generate a force. The type of contraction can be eccentric or concentric. In concentric contractions, the muscles tend to shorten [95]. During this time, the tension within the muscle is constant [96]. In eccentric concentration, the muscles elongate as they face resistance. When the external force is greater than what can be generated by the muscle, it goes through a forced lengthening. Many normal activities, like walking, are considered eccentric and are thus a popular area of study [97]. Many muscular injuries are also associated with eccentric contractions [98].

2.11.2 Isometric

In isometric contractions, the muscle length does not change, however, the energy and tension within the muscles keep fluctuating, and this allows the production of a force. Typically, isometric contractions are observed when there is an action performed towards a fixed object without any resultant movement.

2.11.3 Application of Isotonic and Isometric Contractions

The study of muscular fatigue and neuromuscular diseases can be performed through EMG signals that are obtained through isotonic conditions [99, 100]. Many assistive robotic systems, exoskeletons, and lower-limb orthoses, are designed based on isotonic and isometric contractions [101, 102]. A wide variety of techniques have been described in various literature that focus on applying these signals for the purpose of controlling a rehabilitative robotic device [103-106].

2.11.4 Electromyography (EMG)

The EMG is a complex signal [107] and has many dependencies on the physiological and anatomical properties of the underlying muscles. The most convenient approach to obtaining EMG signals is through the surface of the skin. Thus, it is also known as sEMG (surface EMG). However, the signals obtained through this method also tend to have a lot of noise due to inputs from other neighbouring motor units. A wide variety of signal processing and feature extraction techniques have to be utilized to obtain a detailed analysis of the data obtained.

In our brains, the nerves conduct electrical potentials to convey information. A similar phenomenon is observed with muscles. The motor unit action potential (MUAP) is the combined effect of the various muscle fibres associated with a single motor unit. It can be modelled using this formula:

$$x(n) = \sum_{r=h(r)0}^{N-1} h(r)e(n-r) + w(n) \quad \text{Eq. 2.1}$$

In the formula, N is the number of motor units that are firing, $w(n)$ represents the white noise, $h(r)$ represents MUAP, $e(n)$ is the processed point and $x(n)$ is the modelled EMG signal. Generally, many integrated sensors may perform some basic signal processing techniques using some HW. These include amplification, filtering to remove artefacts and amplitude detection. They may also provide the raw data to allow custom signal processing and feature extraction techniques. The general steps to the use of EMG signals is shown in the Figure 2.5.

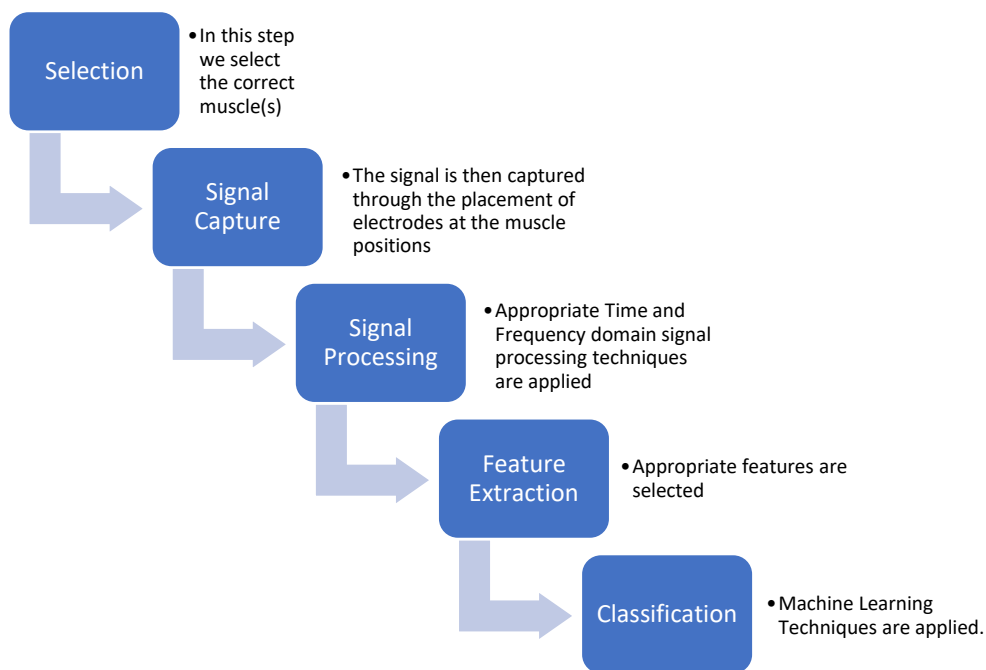


Figure 2.5: Block Diagram for the use of EMG signals

2.11.5 Signal Processing

Signal Processing is critical to ensure that we are able to obtain useful information from the collected data. Various techniques have been explored and the results of the processed signals are then fed to different controllers to perform the required actuation to achieve the therapeutic effect.

One of the first steps in data collection is the process of windowing. Data of varying lengths has been shown to have a direct effect on the final classification error [108]. When the window length was smaller than 128ms, it was shown to have a high level of bias and the features also had a high variance [109]. In similar studies [104, 110], it was shown that if the

segment length was increased from a value of 125ms up to 500ms, the classification accuracy also increased. Increasing the window allows us to capture more information about the signal which leads to a smaller bias and variance in the extracted features. This also helps in the real-time response of rehabilitative systems [111] which should be within 300ms [112].

We can use either adjacent windowing or overlapping windowing technique. The choice is dependent on the amount of work that needs to be done within that time interval. For longer time intervals, some overlap may be necessary to ensure that we are able to achieve the required real-time response.

We will now review some commonly used signal processing techniques used for physiological data.

- **Frequency-Selective Digital Filters**

Filters like low-pass, high-pass, band-pass and band-stop are commonly used to focus the attention on specific frequency bands to perform the necessary detailed analysis. In [113], a band-pass filter with a range of 20Hz – 500Hz was used to remove motion artefacts. In the study of signals from lower-limbs, different filtering techniques were employed. In [114], a 6Hz Low-Pass Filter was applied. [115] made use of a low-pass filter of 6th order and a cut-off frequency of 5Hz, while [116] used a 5Hz to 500 Hz band-pass filter. To reduce the impact of motion artefacts, [113] used a band-pass filter with the frequency range of 20Hz to 500Hz. In most of these observations, noise was still present and very difficult to be totally removed [117].

- **Adaptive Filtering**

In Adaptive Filtering (AF), we perform iterative analysis to establish a relationship between two signals. An advantage of AF is that it is able to modify the signal's features while performing filtering even if there is some overlap of the spectrum [118]. In order to remove ECG artefact, [119] evaluated the use of AF techniques and found it to be effective. In the application of AF, an adaptive whitening filter was used in [120].

- **Common Average Referencing (CAR)**

In CAR, spatial filters are applied across several electrodes. This allows us to compute the mean across the various channels. With this approach, the overall Signal-to-Noise ratio (SNR) improves and it can lead to better classification results. In [121], a 60% increase in performance was seen with the use of CAR. In [122], an improved classification accuracy was observed with the use of CAR.

- **Surface Laplacian (SL)**

Surface Laplacian is a technique that helps in improving the SNR and identifying effective signal sources. [123] found that SL was more effective in removing artefact noise compared with CAR in multi-electrode systems. In multi-channel sEMG recordings, [124] was able to distinguish data from the various sensors using the SL approach.

2.11.6 Feature Extraction

Feature extraction is critical for a rehabilitative system to extract meaningful information from the signals obtained. It involves transforming raw sensor data into feature vectors that can then be used in the Machine Learning layer. These features are generally divided into three main categories. They are Frequency Domain (FD), Time Domain (TD) and Time-Frequency Domain (TFD) features. For TD, the features can be related to the amplitude variations of the signal over time, and also the time-correlated parameters between adjacent signals or from signals obtained from various electrodes. Feature extraction within the Time-Domain allows the computational complexity of the system to be kept low as no further transformation is needed. For FD features, the most common one is the computation of the Power Spectral Density (PSD). The PSD value can be used to obtain features based on different frequency bands that the sensor may exhibit. The TFD approach is able to provide us with very useful information on how the frequency component of the signal varies with time. This is well-suited for non-stationary signals.

Time Domain (TD) Features

We will first look at some commonly-used TD features proposed by [125].

- **Mean Absolute Value (MAV)**

To obtain the MAV, we sum the data points of signal within the region of interest. We then divide this sum by the total number of samples within that region. This is given as

$$MAV = \frac{1}{N} \sum_{n=1}^N |x_n| \quad \text{Eq. 2.2}$$

where x_n represents the data points and N is the number of samples.

- **Mean Absolute Deviation (MAD)**

MAD is an extension to MAV. After calculating the MAV, we obtain the absolute difference between each data point and the MAV and sum them up. This sum is then divided by the total number of samples within the data region.

$$MAD = \frac{1}{N} \sum_{n=1}^N |MAV - x_n| \quad \text{Eq. 2.3}$$

- **Mean Absolute Value Slope (MAVSLP)**

As a modification to MAV, the differences between adjacent values are computed.

$$MAVSLP_i = MAV_i - MAV_{i-1} \quad \text{Eq. 2.4}$$

- **Simple Square Integral (SSI)**

Simple Square Integral uses the energy of the sEMG signal as a feature.

$$SSI = \sum_{n=1}^N |x_n|^2 \quad \text{Eq. 2.5}$$

- **Root Mean Square (RMS)**

To calculate the RMS, the arithmetic mean of the square of all the signals is computed.

$$RMS = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2} \quad \text{Eq. 2.6}$$

- **Variance (VAR)**

To compute the variance, we compute the squared distances between the mean and each signal, and then divide it by the mean. In the equation \bar{x} represents the mean of the signal.

$$VAR = \frac{\sum (x_i - \bar{x})^2}{N - 1} \quad \text{Eq. 2.7}$$

- **Standard Deviation (SD)**

The square-root of the variance gives us the standard deviation.

$$SD = \sqrt{\frac{\sum (x_i - \bar{x})^2}{N}} \quad \text{Eq. 2.8}$$

- **Waveform Length (WL)**

WL is the cumulative length of the waveform over the time duration of interest. It is related to the time, frequency and amplitude of the signal.

$$WL = \sum_{n=1}^{N-1} |x_{n+1} - x_n| \quad \text{Eq. 2.9}$$

- **Zero Crossing (ZC)**

ZC refers to count of the instances where the amplitude of the signal crosses over the zero axis. A threshold condition is introduced in the computation that allows the signal to abstain

from any background interference. This feature also embodies from frequency domain elements.

$$ZC = \sum_{n=1}^{N-1} [sgn(x_n \cdot x_{n+1}) \cap |x_n - x_{n+1}| \geq threshold]$$

Eq. 2.10

$$sgn(x) = \begin{cases} 1, & \text{if } x \geq threshold \\ 0, & \text{otherwise} \end{cases}$$

- **Slope Sign Change (SSC)**

This is quite similar to the idea of ZC. For 3 consecutive segments, the number of times there is a change between the negative and positive slope is measured.

$$SSC = \sum_{n=2}^{N-1} [f[(x_n - x_{n-1}) \cdot (x_n - x_{n+1})]]$$

Eq. 2.11

- **Skew (SKEW)**

This is a measure of how the probability distribution of the signal values is asymmetric. It is also commonly known as the third standardized moment. σ refers to the standard deviation of the signal.

$$\tilde{\mu}_3 = E \left[\left(\frac{X - \bar{x}}{\sigma} \right)^3 \right]$$

Eq.2.12

- **Kurtosis (KURT)**

This measures the extent to which the effect of outliers affects the tails of the distribution.

$$KURT(X) = E \left[\left(\frac{X - \bar{x}}{\sigma} \right)^4 \right]$$

Eq. 2.13

In [106], it was observed that when the sampled data was 200ms, it took around 10ms to perform the feature extraction. A combination of the above-mentioned TD signals were used in [126, 127] to detect the movement of the hand and interpret EMG signals while performing

specific hand-lifting tasks with load. In those papers, it was found that the use of RMS and MAV, together with SD, was an effective feature vector.

In another study, fuzzy approximate entropy (fApEn) [128] was used to measure the dynamics of EMG signals of patients. In [129], Skewness (SKEW) and Kurtosis (KURT) were used with promising results. In [111], moving ApEn was found to be effective in classifying the various stages of muscle contraction. RMS was found to be effective in capturing features from multi-channel EMG sensors in [113]. In another study, [115], RMS was found to be more effective than other TD features for electrode selection. In [130], several TD features were studied and it was concluded that each had their own strengths and weaknesses. The next step in feature selection was considered critical to the overall effectiveness of such systems. Table 2.6 shows a summary of the TD features and the classification accuracies achieved.

Table 2.6: Quantitative summary of TD features

Reference	Features	Classification Accuracy
[126]	Max Amplitude	% Error = 4.1% - 26.2%
	Standard Deviation	% Error = 9.3% - 14.8%
	Root Mean Square	% Error = 4.1% - 26.2%
[116]	Waveform Length	% Error = 3% - 8%
	Standard Deviation	% Error = 2% - 6%
	Mean Absolute Value	% Error = 4% - 6%
[105]	Mean Absolute Value	86.49 ± 9.6 %
	Simple Square Integral	78.63 ± 9.1 %
	Waveform Length	88.72 ± 7.2 %
	Zero Crossing	86.93 ± 8.1 %
	Slope Sign Change	88.31 ± 7.7 %
	Root Mean Square	86.21 ± 9.4 %
	Variance	78.42 ± 9.2 %
[131]	Skew	87.84% - 97.66%
[132]	Slope Sign +	
	Waveform Length +	85.6 ± 4.8 %

Auto-regressive Coefficients		
[133]	Moving Average + Waveform Length + Zero Crossing + slope Sign Change	95 ± 4%
	Root-Mean-Square + Auto- Regression	96 ± 3%

Frequency Domain (FD) Features

There are a wide range of FD features that can be used for EMG signals and physiological signals in general. In the case of rehabilitation, they have been used quite extensively in the areas of muscle fatigue assessment and motor unit analysis [96]. The mean power frequency (MPF) of EMG signals is able to provide information regarding the complex signal changes generated by muscles. For stroke patients, it has been shown that the MPF is lower for their paretic muscles as compared to the contralateral muscles.

Signals captured through muscle contractions can be characterized through the use of the median power frequency (MNP), mean frequency (MNF) and power spectral density (PSD). From these, you can compute the mean, median from the power spectrum or amplitude spectrum, and use them as features. This is summarized in Table 2.7.

Table 2.7: Quantitative summary of FD features

Reference	Features	Accuracy
[115]	Mean Frequency, Median Frequency, Signal-to-Noise Ratio	Repeatability: 75% Repeatability:75% Repeatability:64.29%
[116]	Spectral Energy	Classification: 91.3 ± 3.5%
[105]	Peak Frequency Spectral Moment	Classification: 44.63 ± 7.2% Classification: 80.29 ± 9.1%

Mean Power Frequency	Classification: $78.31 \pm 9.3\%$
Total Power	Classification: $78.61 \pm 9.1\%$
Frequency Ratio	Classification: $69.81 \pm 8.9\%$
Variance of Central Frequency	Classification: $16.93 \pm 2.6\%$
Power Spectrum Ratio	Classification: $49.72 \pm 8.9\%$

Time-Frequency Domain (TFD) Features

Studies that have explored both TD and FD features seem to suggest mixed-results with no clear conclusion on which approach would be better [100, 101, 105, 106]. As an alternative, TFD features have been proposed as a means to overcome the shortcomings of TD features due to their stationary nature. In [134], it was found that the use of wavelet packet transform features yielded a lower classification error of 6.25%, when compared to the use of TD features which had an error 9.25%. TFD features like short-time fourier-transform (STFT) and wavelet transform (WT) are able to localize the time-frequency energy and are suitable for capturing the information in EMG signals [135]. One of the challenges of TFD features seems to be the high dimensionality together with the high-resolution of the feature vectors [117]. Thus, it is critical to perform a reduction in dimensionality while maintain the ability to discriminate the features [101, 111]. To reduce the dimensionality, we can perform feature projection, where the original features are projected to a new feature space that is smaller. In feature selection, we look at the optimal combination of features to form a new set, according to a rule-base. The types of TFD features are as shown here.

- **Continuous Wavelet Transform (CWT)**

The Fourier Transform functions by breaking up a signal into a combination of sines and cosines that are infinite. This causes it to lose its time-localization data. Wavelets represent minute oscillation that have a high localization representation in time. The CWT allows us to decompose a signal into wavelets which gives us a good representation of the time-frequency localization.

The continuous wavelet transform of a function $x(t)$ at a scale ($a > 0$) and translational value $b \in \mathbb{R}$ is expressed by the following integral

$$X_w(a, b) = \frac{1}{|a|^{1/2}} \int_{-\infty}^{+\infty} x(t) \bar{\psi}\left(\frac{t-b}{a}\right) dt \quad \text{Eq. 2.14}$$

where $\psi(t)$ is a continuous function in the time-frequency domain and it is referred to as the mother wavelet.

- **Discrete Wavelet Transform (DWT)**

In DWT, the wavelets are sampled discretely. They offer the same advantage over the Fourier Transform by offering time-frequency localization information. It has been used extensively in a wide range of applications. Two popular domains where it has been widely used are in data compression and filtering. A wide range of wavelets have been proposed and used and the most common ones are the Haar wavelet and the Daubechies wavelet.

We can compute the DWT of a signal x by passing it through a filter series. At the first stage, the signal is convoluted with an impulse response g and passed through a low pass filter.

$$y[n] = (x * g)[n] = \sum_{k=-\infty}^{\infty} x[k]g[n-k] \quad \text{Eq. 2.15}$$

A high-pass filter h is also used to decompose the signal simultaneously. We then obtain two outputs simultaneously. They are the output from the high-pass filter, also known as the detail coefficients. We also have the output from the low-pass filter, the approximation coefficients.

We are then able to perform subsampling by 2 by passing the signal through new filters g and h with the cut-off frequency being halved as compared to the previous one.

$$y_{low}[n] = \sum_{k=-\infty}^{\infty} x[k]g[2n-k] \quad \text{Eq. 2.16}$$

$$y_{high}[n] = \sum_{k=-\infty}^{\infty} x[k]h[2n-k] \quad \text{Eq. 2.17}$$

This decomposition technique is continuously repeated to allow us to have a higher frequency resolution, and this is represented using a binary tree that shows the time-frequency localization. This binary tree, as shown in Figure 2.6, is also referred to as the filter bank.

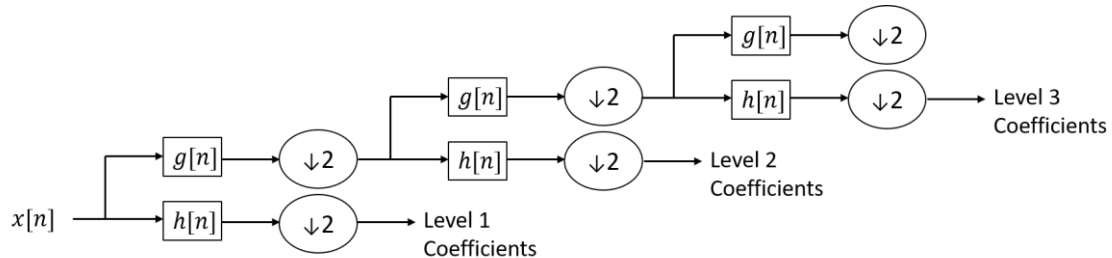


Fig 2.6: Three-Level Filter Bank

Due to the nature of the decomposition process, we need to ensure that the signal is a multiple of 2^n where n is the number of levels. If we have a signal with 32 samples and the range is from 0 to f_n , with three levels of decomposition, we will generate 4 output scales. This can be seen in Figure 2.7.

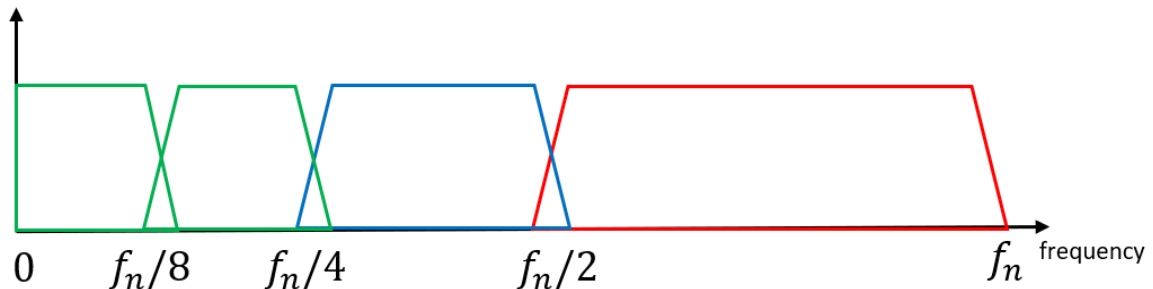


Fig 2.7: DWT Frequency Domain Representation

2.11.7 EEG-based Approach

Over the years, we have seen advances in the brain science domain that has allowed the development of many brain-computer interface (BCI) devices. Many of these devices are targeted towards being applied in the area of neurorehabilitation. This aims to assist those with physical challenges that may be physical, neural or both. Such systems have extensive signal processing and machine learning techniques due to the complex nature of the brain wave signals.

The idea of developing a robotic arm using brain signals was first introduced in [136]. The aim of BCI systems was to capture the activity in the brain and use it to interpret the intention of the person [137]. Such BCI systems can be feedforward-based where the intention is decoded and translated into an action. The feedback-based approach aims to capitalize on the neural plasticity nature of the human brain to aid in the rehabilitation process [138-142].

A wide variety of invasive and non-invasive techniques have been utilized over the years, with the non-invasive ones being the more popular and widely accepted one. These include magnetoencephalography (MEG), functional magnetic resonance imaging (fMRI), electroencephalography (EEG) and near-infrared spectroscopy (NIRS). The ability to non-invasively measure brain signals in a portable and inexpensive device allowed EEG to be preferred signal acquisition method for many BCI devices. Though EEG-based devices have had many application, they have also shown great promise in the area of assistive robotic therapy [143]. Those who face any physical challenges, either through muscular or through neurological deficits can utilize EEG signals to control external devices [144].

BCI-based devices have to be designed with a specific protocol based on the desired inputs and outputs. Experiments are then carried out pertaining to the protocol to capture the acquired data which is then analysed.

Motor-Imagery (MI) Based BCI

In motor-imagery, an individual is tasked with imagining a particular movement. In [145], it was established that the brain activity during an imagined tasks highly correlates with the brain activity of the actual task. This technique involves two main approaches, the sensorimotor rhythm (SMR) and the imagined body kinematics (IBK).

- Sensorimotor Rhythms (SMR)

In SMR, the imagined movement involves that of body parts like the tongue, hands and feet, and this translates into changes in the brain activity [146]. This translates to a phenomenon known as event-related desynchronization (ERD) and event-related synchronization (ERS). This is observed in the mu (8 – 12 Hz) frequency range and the beta (18 - 26Hz) frequency

range. They are most prominently observed in the C3 and C4 electrode locations which are located in the area of the sensorimotor cortex [147]. A wide range of signal processing and classification strategies have been attempted to use SMR for practical BCI applications. In [148], a strategy to control a cursor in two-dimensions was demonstrated using SMR. A variety of classification methods were investigated for right/left hand imagery in [149], and SVM proved to generate the best results. A summary of the research done with the use of SMR is shown below in Table 2.8.

Table 2.8: Motor Imagery-based BCI Systems

Task	Feature	Classification	Reference
1D Cursor Control	Mu rhythm (8 – 12 Hz) amplitude	N/A	[150]
2D Cursor Control	FFT & mu rhythm amplitude	Linear Regression	[151]
Prosthetic Hand Control	Peak mu band power	Logistic Regression	[152]
Rehabilitation Robot Control	CSP features	N/A	[153]
Rehabilitation Robot Control	Time-Frequency Power in alpha, theta and beta bands	LDA	[154]

- Imagined Body Kinematics (IBK)

Though SMR has been used extensively in many BCI applications, one of its main drawbacks is that it lacks intuitive and natural control [155]. The features extracted from SMR can give us general information on the body part that we intend to control, however detailed parameters like the velocity and position are missing. IBK originally had its roots in invasive BCI [156, 157] but non-invasive techniques have found that it is possible to extract meaningful information from low-frequency signals (less than 2Hz).

It has been shown that the EEG signals in the low-frequency range ($< 2\text{Hz}$) that are located in the sensorimotor cortex region exhibit kinematic data. In [158], this technique was used to decode and classify wrist extension and rotation at varying speeds. In [159] and [160], the authors were able to extract and classify two-dimensional and three-dimensional data respectively.

- Steady-State visual evoked potential (SSVEP)

In SSVEP, subjects are shown a flickering target. Different targets may flicker at different rates. It has been observed that there is a strong correlation between the frequency of flickering for the targets and the response observed in the EEG signals. Though it has proved to be a viable system for some applications, it is not suitable for those with visual impairment, and can also cause fatigue for users.

- Error-related potential (ErrP)

In error-related potential, we are able to detect whenever the resultant action has a mismatch with the intention. For example, if a user wants to move a cursor in a particular direction, but the movement is in the opposite direction, then an error signal can be observed. This was found to be most pronounced within a 200 to 700ms window around the central and frontal lobes. Such techniques were tested to be effective in some studies [161].

- Hybrid Approach

A hybrid system consists of one or more physiological parameters where at least one of the input is EEG. It can be a case where EEG signals are decoded and processed with a variety of techniques and the combined outputs of these subsystems can control a BCI application. This idea can be extended so that it can be combined with other physiological signals to create a more robust system for BCI applications. A summary of these various techniques by researchers is shown in Table 2.9.

Table 2.9: Applications of BCI-based systems

Task	Hybrid Paradigm	Reference
Assistive Robotic Arm	P300 & SSVEP	[162]
2D cursor task	Mu/Beta & P300	[163]
Robotic Grasp Motion	SSVEP, Mu	[164]
Quadcopter control	Mu/Beta & CCA	[165]
Point and Select	Mu/Beta & CCA	[166]

A more detailed review and study can be found in these articles: [167-169]

2.12 Classification & Machine Learning (ML) Models

We will first review some of the commonly used Classification and ML techniques before looking at their application in this domain.

- Multilayer Perceptron (MLP)

The MLP is used quite loosely to refer to feedforward Artificial Neural Networks (ANNs) in general. It consists of an input layer, an in-between hidden layer, and an output layer. The MLP is a model that utilizes supervised learning using a common backpropagation technique [170]. To closely resemble the natural behaviour of biological neurons, a non-linear activation function is used. Though sigmoids have been commonly used in the past, in more recent times, the rectified linear unit (ReLU) has gained more acceptance and popularity. A simplified model is shown in the Figure 2.8.

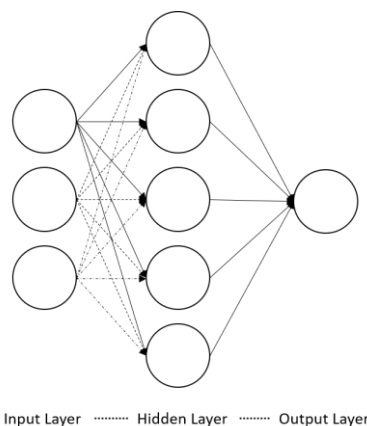


Fig 2.8: Multilayer Perceptron

- Support Vector Machine

SVM's are a popular prediction model that is based upon statistical learning [171]. It is a supervised learning model, where, given a set of training data that is already labelled into categories, the algorithm develops a model that is able to assign new data samples into the appropriate category. The algorithm aims to maximize the gap between the categories to enable a higher level of classification accuracy. An example is shown in Figure 2.9.

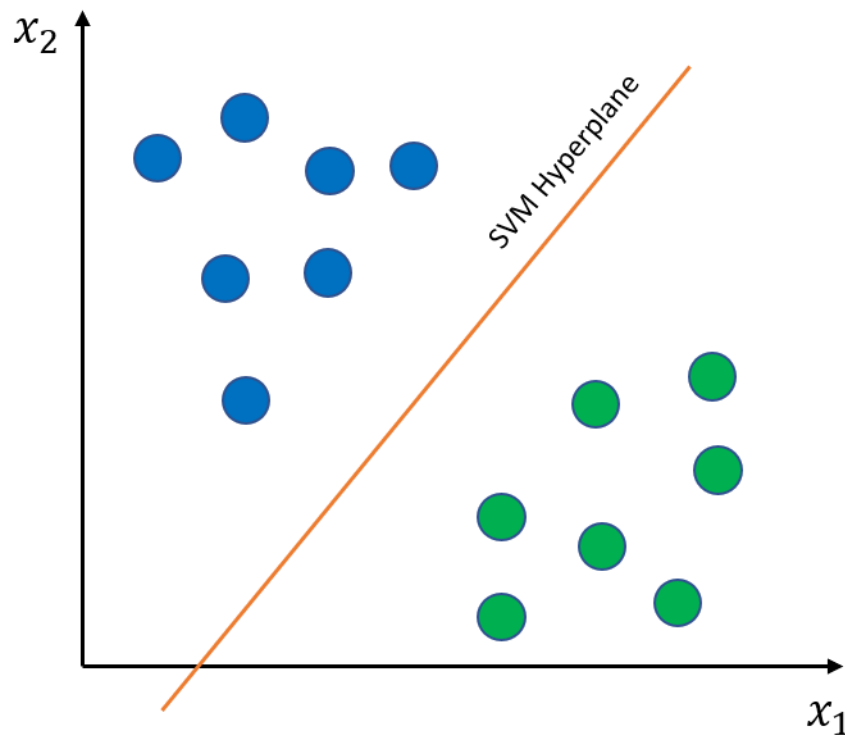


Fig 2.9: SVM Hyperplane

- **Linear Discriminant Analysis (LDA)**

Linear Discriminant Analysis (LDA), also commonly known as the Discriminant Function Analysis, is a generalized form of the Fisher's Linear Discriminant. The aim of this technique is to find a set of features in a linear combination that is able to separate classes of objects. It is closely linked to another technique, Principal Component Analysis (PCA).

In [131], the objective was to reduce the distance that was observed amongst the samples within the same class, while maximizing the distance that was observed amongst the center's of the different classes. The Orthogonal Fuzzy Neighbourhood Discriminant Analysis (OFNDA)

was proposed as a means to achieve this objective. In [133], the Support Vector Machine (SVM) was compared against the Linear Discriminant Analysis (LDA) and the Multilayer Perceptron (MLP). It was shown that the SVM was able to yield higher classification accuracies as compared to the other two models. In [105], a comprehensive study was done to compare the performance of the following techniques: Random Forests (RF), Linear Discriminant Analysis (LDA), Decision Tree (DT), Support Vector Machine (SVM), k-Nearest Neighbour (kNN), Quadratic Discriminant Analysis (QDA) and Multi-Layer Perceptron Neural Networks (MLP-NN). It was shown that LDA provided a classification accuracy of 98.87% by utilizing Time-Domain features. In [172], the authors used MLP to on TFD features for forearm motions to achieve a 99% classification accuracy.

Table 2.10: Machine Learning Techniques and their Results

Reference	Model	Classification Accuracy
[131]	Orthogonal Fuzzy Neighbourhood Discriminant Analysis	87.84% - 97.66%
[133]	Support Vector Machine	95.5 ± 3.8%
	Linear Discriminant Analysis	94.5 ± 4.9%
	Multilayer Perceptron	89.5 ± 4.8%
[105]	Linear Discriminant Analysis	98.87 ± 0.8%
[172]	Multilayer Perceptron	99%

2.13 Deep Learning

In recent times, deep-learning based techniques have been widely used in many machine learning applications that involve physiological signals like EMG and EEG. We will first review some of the basics of this approach.

A. Basics of Deep Learning

The main idea behind deep learning is that it uses multiple hidden networks to learn high-level characterizations of input data. The network grows deeper as the number of hidden layers increases. This idea originated in Hinton et. al. [36] who utilized unsupervised greedy training methods to train deep belief networks.

Convolutional Neural Network (CNN)

The CNN was first proposed in [173] by LeCun et.al, to classify handwritten digits. Over the years, the CNN has been adapted to create many other variants, such as, VGGNet, LeNet, and AlexNet. CNN's generally consist of two stages, convolution and pooling. Using a range of filters in the convolution stage, several features are first extracted from an image automatically. In the pooling layer, the most significant features of the original image are kept while decreasing the number of features. This process is repeated a few times and the extracted features are fed into fully-connected layers and an output layer to perform the classification. While a MLP has fully connected neurons, with each neuron connecting with all the neurons in the next layer, the CNN only has local connections between the neurons and adjacent layers.

Recurrent Neural network (RNN)

Originally called the Elman Network, it was presented in 1990 by Elman et.al. [174]. Generally, neural networks do not consider the temporal relationship between individual data points and tend to fit the input data into individual labels. The purpose of the RNN was to represent this temporal information within a sequence. Its structure is similar to a MLP, where there is an input layer, output layer and hidden layers. The key difference is that the nodes in the hidden layer are connected with earlier nodes. Thus, the current input, together with the state of the previous hidden layer is sent to the current hidden layer. This allows the network to learn the relationship between the input sequences.

RNN's have an issue with the amount of information it can retain due to the issues of gradients exploding or disappearing. The Long-Short Term Memory [175] was proposed as a solution to this issue. The LSTM block has a forget gate which allows a means to control how much of the preceding information should be retained or forgotten.

Auto-Encoder (AE)

The Auto-Encoder (AE) was first proposed in 1987, and its main benefit was its ability to perform well with unsupervised pre-training [176]. AE has shown promise in areas such as

fault-detection, medical imaging and other applications. It consists of an input encoder and an output decoder. Using appropriate loss functions, the network aims to minimize the differences between the input data and output labels. At times, AE is also used as a data dimension reduction tool.

Deep-Belief Network (DBN)

The Restricted Boltzmann Machine (RBM) is the basic component of the DBN. The RBM belongs to a special type of Markov Random Field, which consists of a visible layer and a hidden layer. Similar to the AE, without the use of a predefined label, the RBM is able to reconstruct data using an unsupervised machine learning method. Cascading multiple RBM's creates a DBN, with each layer being trained using a greedy algorithm. The DBN attempts to gain a better understanding of the data without any prior knowledge of the tasks carried out.

A summary of the use of Deep Learning techniques and their results is shown in Table 2.11.

Table 2.11: Comparison of Deep Learning Techniques

Dataset	Network	Accuracy	Ref
NinaPro DB2	Two Stream CNN	75% - 84%	[177]
28 Participants	DBN	88% - 89.29%	[178]
5 Conditions	LSTM	87% - 94%	[179]
NinaPro DB3	CNN	73.31%	[180]

2.14 Edge Computing

Research into novel signal processing and machine learning techniques are generally conducted using high-end computing platforms with its accompanying software tools. However, to deploy such algorithms and models onto devices for daily use, they need to be ported to embedded platforms. These platforms need to be able to perform complex computations on the end device, and hence such a technique is also known as “Edge Computing”.

There are many advantages of performing machine learning "at the edge". In particular, lower latency and reliability, as the information is processed close to the sensors, without needing to transmit it to the cloud. The communication with the cloud would usually require milliseconds of latency, compared to the microsecond that can be achieved with a low power microcontroller. Improved data protection, as the data are not transferred to another computer through the Internet or by other wireless or wired communication. No wireless or wired communication is needed — or if it is needed — a much lower bandwidth is required, as the processed information is an order of magnitude less than the raw data. For example, several kilobytes would be needed to send an image of an apple, but after the image has been classified as an apple, the fact that it's an apple could be communicated to the cloud in just one byte. The Machine Learning can be processed in a low-power microcontroller, which consumes only milliwatts of power. Moving Machine Learning from the cloud, to the edge, and to the extreme edge, working with microcontrollers, means it can run inside small, low-energy, battery-operated devices, including wearables such as earbuds. However, despite the benefits, this requires us to work with processors that have vastly lower computation resources. Rather than several trillions of operations per second, a processor such as the Arm Cortex- M4 may only perform several million operations. What's more, the memory is also significantly reduced from several terabytes to a maximum of a few megabytes in the best case. It's important to note that in supervised Machine Learning the inference step might require millions of operations that are mainly multiplications and additions. So, memory is necessary to store the data, the parameters of learning, and the results of the run-time calculations. Running these operations on a high-performance embedded processor, such as the Cortex-M4, minimizes the time for inference as much as possible by exploiting the core digital signal processor extension and the floating-point unit to run the model in the most efficient way. The main goal of Machine Learning on microcontrollers is to design a deployable and optimized solution on low-power resource-constrained hardware.

Review of Edge Computing

Internet of Things Devices, mobile platforms, embedded solutions, etc. rely on Microcontrollers, which are devices with resource-constraints. Microcontrollers are single-chip solutions which integrate the Central Processing Unit (CPU), Memory and Power subsystems, together with other peripheral blocks into a single package. Some of the popular

microcontrollers and their basic specifications are as shown in Table 2.12.

Table 2.12: Comparison of Microcontrollers

Processor	Name	Clock	Memory
ATMega328P	Arduino Uno	16 MHz	Flash: 32KB, RAM: 2KB
ATMega2560	Arduino Mega	16 MHz	Flash: 256KB, RAM: 8KB
ARM Cortex M3	STM32F2	120 MHz	Flash: 2 MB, RAM: 128KB
ARM Cortex M4	STM32F4	180 MHz	Flash: 2 MB, RAM: 384KB
ARM Cortex A53	RPi 3B	1.2 GHz	Flash: SD Card, RAM: 1GB

Machine-Learning algorithms are computationally intensive and General-Purpose Microcontrollers alone are not efficient in handling such tasks. Dedicated Hardware Accelerators can be used to supplement the Microcontroller to provide it with the necessary computing power that is needed. These include devices like the ARM Ethos NPU, BeagleBone AI, Intel Movidus NCS, NVIDIA Jetson NANO and many others. These hardware accelerators are computationally efficient, but not optimized for power consumption.

TinyML is a machine learning approach that focuses on integrating optimized and compressed machine learning models to be well suited for deployment in low-power microcontrollers. It boasts a range of advantages ranging from data security, reduced latency, energy efficiency, and many others. Table 2.13 shows the popular TinyML frameworks and their details.

Table 2.13: Comparison of TinyML Framework

TinyML Framework	Algorithm	Platform
ARM-NN	Neural Networks	ARM Cortex-M ARM Cortex-A ARM Mali
TensorFlow Lite	Neural Networks	ARM Cortex-M
CMSIS-NN	Neural Networks	ARM Cortex-M
STM32Cube AI	Neural Networks	STM32
MicroMLGen	SVM	ESP32 ESP8266

In [181], research was carried out to investigate the ability for a network to auto-configure its radio and IP stack by utilizing ML techniques. Using the IBM PPC and a ARMv7 processor, a the SVM algorithm was implemented. It was found that the deployment of the SVM was efficient only with specific optimizations of the algorithm. These include tradeoff between execution speed and accuracy by foregoing floating point operations. This reduced the overall accuracy of the system, though execution saw an improvement. The authors in [182] used the FRDM-K64F board and deployed a Gaussian Mixture Model (GMM) onto it to evaluate the ability to rform real-time analysis of sensor data. The NXP ISF (Intelligent Sensor Framework) was used to manage the sensor data. The GMM algorithm was inherently computationally intensive with extensive matrix and vector operations. It was found that by computing sensor data using fixed-point representation, the computational power requirements were reduced at the expense of accuracy. In [183], the research investigated the deployment of Deep Neural Networks across a range of IoT platforms with a focus on reduced power consumption. A System-on-Chip (SoC) hardware accelerator is used to infer the trained model. Several optimization techniques had to be employed, such as reducing weights and biases from 32-bit to 16-bit, and improving the quantization process. The results demonstrated that it was possible to implement a deep learning model on SoC based devices. The authors in [184] integrated ML algorithms onto wearable devices to perform all the required processing on the device. This included sampling, feature extraction and performing the classification on a SVM model. The objective was to avoid the need for cloud-based computing and perform all operations on the embedded device. It was shown that by performing all processing on the device, the battery life of the device was extended many-fold as compared to performing cloud-based processing. This is achieved through the use of appropriate SoC hardware. In [185], researchers focused on developing a novel technique to employ low precision arithmetic with the aim of optimizing the power consumption of the device. Several number conversion techniques were utilized to reduce the overall computational complexity. The proposed technique showed good accuracy with reduced complexity.

2.15 Summary of Literature Review

From this comprehensive literature review, it is clear that there have been extensive techniques and approaches used by researchers in the area of rehabilitative robotics. These

include the use of different types of sensors, signal processing techniques, as well as machine learning techniques. There still various other techniques that can be explored to design and develop more advanced systems. They should be able to adapt to the user's progress in real-time as well as for them to get qualitative feedback about their progress.

2.16 Scope of Focus of this Research

The aim of this research is to investigate various aspects of designing and developing systems that can aid in the rehabilitation of patients. This research aims to achieve the following outcomes:

- Investigate the use of physiological signals such as EEG and EMG to accurately decode and classify a range of real-world actions.
- Develop novel signal processing techniques that utilizes the unique temporal and spatial relationships between various physiological signals to aid in the rehabilitation.
- Develop a novel inference system that is able to mimic the neuroplasticity of the brain such that it is able to accurately decode and classify motor actions based on the available physiological signals.

Publication:

A Comprehensive Review of Motor Movement Challenges and Rehabilitative Robotics

Smart Health, Volume 29, 2023, 100402, ISSN 2352-6483

Journal Paper

DOI: <https://doi.org/10.1016/j.smhl.2023.100402>

CONTRIBUTORS: Ravi Suppiah; Anurag Sharma; Noori Kim; Khalid Abidi

Chapter 3

Novel EEG Model

3. 1 The EEG Signal

In the Literature Review, we understood the basics of the brain's anatomy and how signal is transmitted through the brain via its neurons. The vast network of neurons generates electrical impulses that transcend throughout the brain network depending on the stimulus or intended action. In this chapter, we will look at the brain structure in more detail and how EEG signals can be deciphered to better understand the intent. It is possible to use basic signal processing and feature extraction methods to build a simple rehabilitative system. However, such a system would be limited in its ability to fully decode variations and details in the user's intent. We will now explore the use of EEG signals and more complex techniques in decoding the intention of the user.

Mapping of The Human Anatomy to the Brain

For many years, the widely accepted mapping of the regions of the brain were based on the findings and publications of Korbinian Brodmann, a German anatomist [186]. The Brodmann areas have been debated extensively and it has gone through changes whenever new data was available. Many of these areas have internal structures that are complex. A topographic map allows us to map parts of the body or at a higher-level of abstraction to a region in the cortex. The mapping that is of significance in this research is that of the somatosensory cortex. We are able to observe electrical signals in the neurons in those areas when there is a corresponding muscular movement.

Over the years, technological improvements such as the functional magnetic resonance imaging (fMRI) allow us to capture blood flow patterns in the brain in real-time. Other

techniques have complemented each other to help us obtain a more detailed analysis of the activities of the brain. The Human Connectome Project [187] allowed us to have a breakthrough in the mapping of the brain and analyse parts of the brain that were missed in other studies.

Capturing EEG Signals

EEG signals can be acquired through a variety of sensor-based devices. There are commercial devices with just one or two sensors, right up to those that incorporate more than a hundred sensors. These devices generally follow the 10-20 electrode placement strategy. Using the 10-20 system, the position of electrodes on the scalp is fixed according to a pre-set convention that has assigned labels to each sensor. With this labelling, we are able to study and compare the data acquired through different experiments, using similar scalp locations.

3.2 Dataset

Acquiring a dataset of physiological signals presents many challenges. In most countries, regulatory approval is needed at both the national and educational/research institute level to conduct human trials. There are also many other administrative matters to be resolved. We also need to fairly compensate the participants for their time and effort. In order not to be delayed in the data collection process, many researchers prefer to use publicly available datasets that have experimental data that matches the research they are doing. A similar approach has been taken with this research thesis.

Selection and Structure of Dataset

The dataset selected for this research is obtained from Luciw, M et al. [188]. In this dataset, a total of 3936 grasp and lift trials were conducted with varying friction and weight. Multiple physiological signals were captured for each trial. These include the 32 channel EEG data, 5 EMG sensor data, and the force/torque of experienced during the action as well as the 3D position of the object. One of the key objectives of collecting this dataset was to investigate the effect of EEG signals in the context of multisensory control systems. A total of twelve participants took part in this data collection exercise and. During all the trials, the participants

went through a different series of object's properties which were not known to them beforehand. In these series, the weight (165, 330 or 660g) and the contact surface (sandpaper, suede, or silk) were the varying parameters. In some trials one of them changed and in some cases, both of them changed. The dataset was captured and neatly organized into separate MATLAB files that allowed researchers to quickly extract out the relevant information depending on their focus. A more detailed explanation of the experimental procedure and the equipment used can be obtained from the reference.

3.3 Methodology

In this section, we look at the various signal processing techniques that were utilized to extract meaningful features from the dataset. Subsequently, we look at the machine learning models and the results obtained from the various techniques.

3.3.1 Data Preparation

The chosen dataset has a wide variety of 'Series' that allows the participants to use different weights with the same surface (weight series) or, use the same surface with different weights (surface series). There are 3 different sets of weight used, 165g, 330g and 660g. In the following analysis, three series from each participant using the 330g weight is extracted. In each series, there are 10 lifts with the expected weight.

Each lift starts off with a rest period of 2s, followed by a forward movement of 2s. The movement is triggered when the participant observes an LED being ON. The weight is held in a fixed position until the LED goes OFF. After that, the weight is lowered and the arm returns to the original position.

In this analysis, the focus is on using the EEG signals to accurately decode the Rest and Active state of the arm. This can be done using the first 4s of data, with the first 2s representing the Rest state and the next 2s representing the Active state.

3.3.2 Signal Processing & Feature Extraction

From the Literature Review, we understand that one of the well-established signal processing techniques was the power band information. The frequency bands of theta, alpha, beta and gamma are computed and analysed.

During the acquisition of the dataset, the first 2 seconds of the data capture are during the rest state of the participant. Subsequently, the indicator LED is switched ON, and the participant reaches forward to grab the weight and lift it up. This step takes another 2 seconds. Looking at the first 4 seconds of the data, we are able to divide it into 2 separate sections that represent both the rest state and the active state. Figure 3.1 shows both the time domain plot and frequency domain plot of this data.

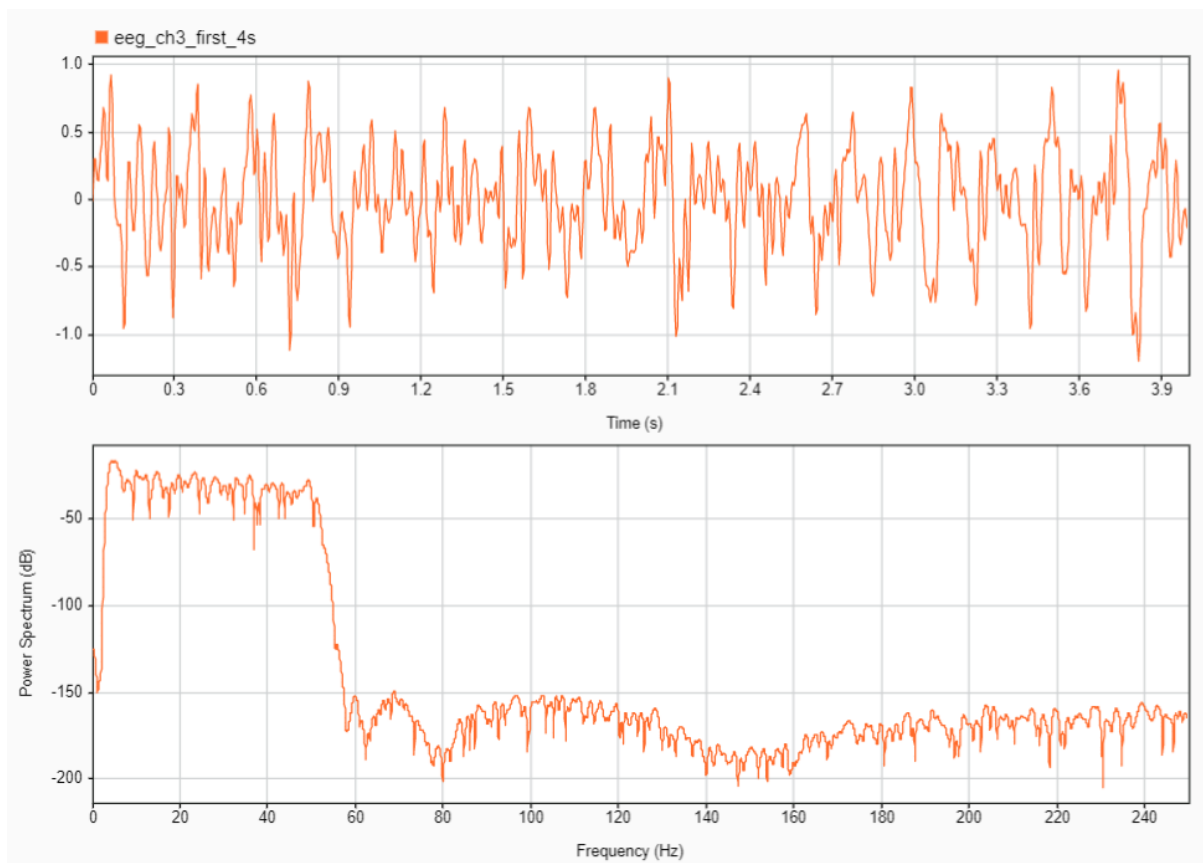


Figure 3.1: Time and Frequency Domain Plots of C3 for the first 4 seconds of data capture.

In the subsequent signal processing techniques, the signal is windowed into 100ms intervals. This gives us 20 data points each, for the Rest and Active states. We have 10 lifts per series and 1 series per participant. That gives us 40 data points each for Rest and Active. For the 12 participants, we have a total of 240 data points for each state and a total of 480 data points.

- **Power Band Analysis**

The signal is band-pass filtered from 0.5Hz to 50Hz. This represents the frequency band of interest that is observable in EEG signals. The signal is then split into TWO 2-second parts. The first 2s data represents the ‘Rest’ state and the next 2s data represents the ‘Active’ state. The various frequency power bands are computed across the rest and active states. This is performed using the periodogram function in MATLAB which is defined as follows:

$$P(f) = \frac{\Delta t}{N} \left| \sum_{n=0}^{N-1} x_n e^{-j2\pi f \Delta t n} \right|^2 \quad -\frac{1}{2} \Delta t < f \leq \frac{1}{2} \Delta t \quad \text{Eq. 3.1}$$

Where Δt , is the sampling interval, N is the number of samples, f is the frequency and x_n is each sample. The following box plot in Figure 3.2 shows the power band values when they are computed across the entire 2s window of both rest and active. It can be observed that both the Alpha and Beta bands exhibit a greater difference in their values when comparing the rest and active states. This applies to both the median and the interquartile range of the data.

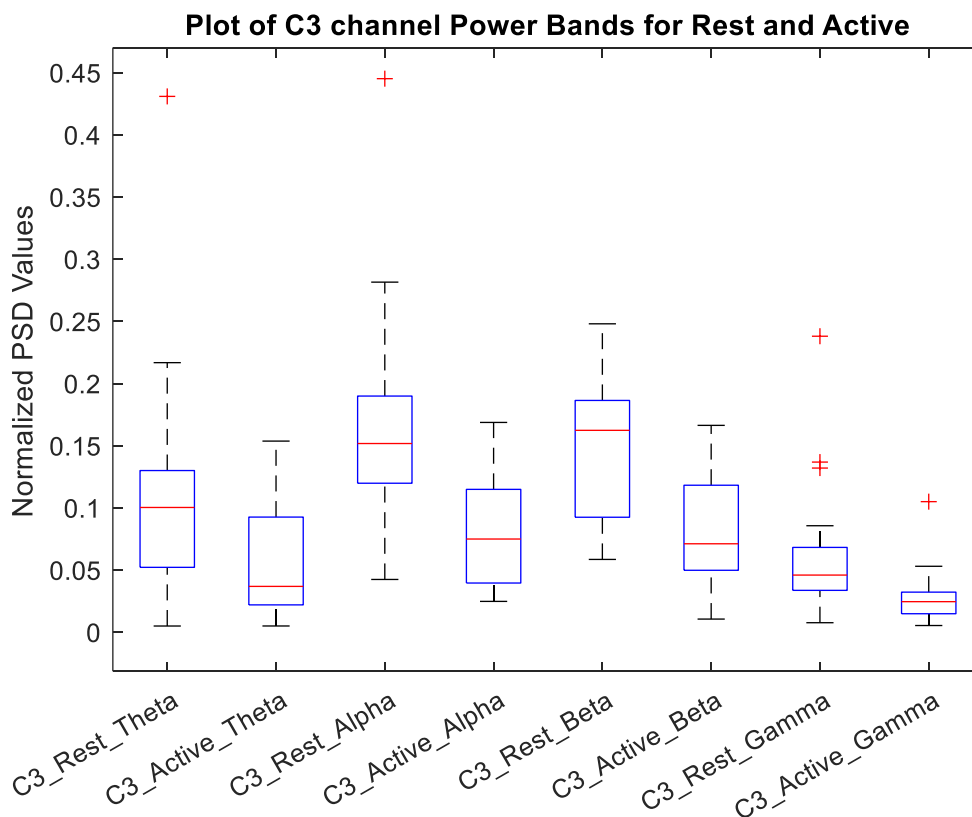


Figure 3.2: Plot of C3 channel Power Bands

The following Scatter Plots in Figure 3.3 show the data and how they are distributed across the different power bands. Each row is represented by PSD values for both rest and active for a particular band.

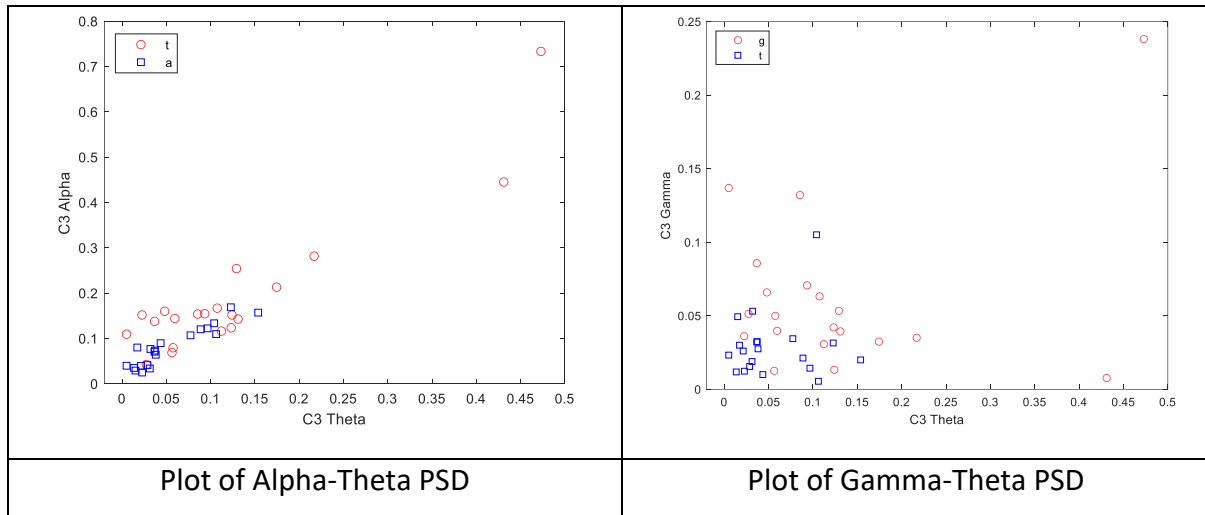


Figure 3.3 Scatterplot of PSD values

It can be observed that the plots between Alpha-Theta PSD exhibit a high degree of correlation. The plots of other PSD pairs are similar to the pattern exhibited by the Gamma-Theta PSD, where they have a high degree of variance. We can now examine the correlation coefficients to better this.

- **Correlation Coefficient Analysis**

The correlation coefficient between two sets of data is a measurement of their linear dependency to each other. The Pearson correlation coefficient can be defined as such:

$$\rho(A, B) = \frac{1}{N - 1} \sum_{i=1}^N \left(\frac{A_i - \mu_A}{\sigma_A} \right) \left(\frac{B_i - \mu_B}{\sigma_B} \right) \quad \text{Eq. 3.2}$$

where σ_A and μ_A relate to the standard deviation and mean of A respectively, and σ_B and μ_B relate to the standard deviation and mean of B. The coefficient values for the scatterplots are as shown in Table 3.1. It can be observed that the correlation coefficients for the Theta-Alpha, Alpha-Beta and Theta-Beta bands and high, which indicate that they behave similarly during the rest and active states.

Table 3.1: Correlation Coefficient Analysis for PSD

Theta - Alpha	0.9834
Alpha - Beta	0.8982
Beta - Gamma	0.2499
Theta-Beta	0.8273
Theta-Gamma	-0.0210
Alpha-Gamma	0.0029

- **ERD/ERS Analysis**

Event-related potentials (ERPs) represent changes in the signal during the onset of some stimuli. These changes can be observed through the analysis of the signal in time, frequency or time-frequency domain. The extent to which these signals may change is dependent on an individual's response and is a variable factor. It has been shown [189, 190], that we can observe frequency specific power band changes in the time-zone surrounding the onset of an event. They reflect the activity that represents an action or imagined action. This phenomenon is called ERD/ERS (Event-Related Desynchronization / Event-Related Synchronization) and it represents the neuronal activity of a node with respect to itself or another node, and across different power bands. The exact behaviour may vary depending on the signals being compared. The approach to compute ERD/ERS is given as such:

1. Bandpass Filtering based on the various trials.
2. Obtaining the PSD values across the trial
3. Compute a reference PSD value based on the 'Rest' period
4. Compute ERD/ERS based the following formula

$$ERD \text{ or } ERS (\%) = \frac{(A - R)}{R} \times 100 \quad \text{Eq. 3.3}$$

where A refers to the PSD of interest and R refers to the reference PSD.

The ERD/ERS plot is obtained for C3 during the first 4s of data capture is shown in Figure 3.4.

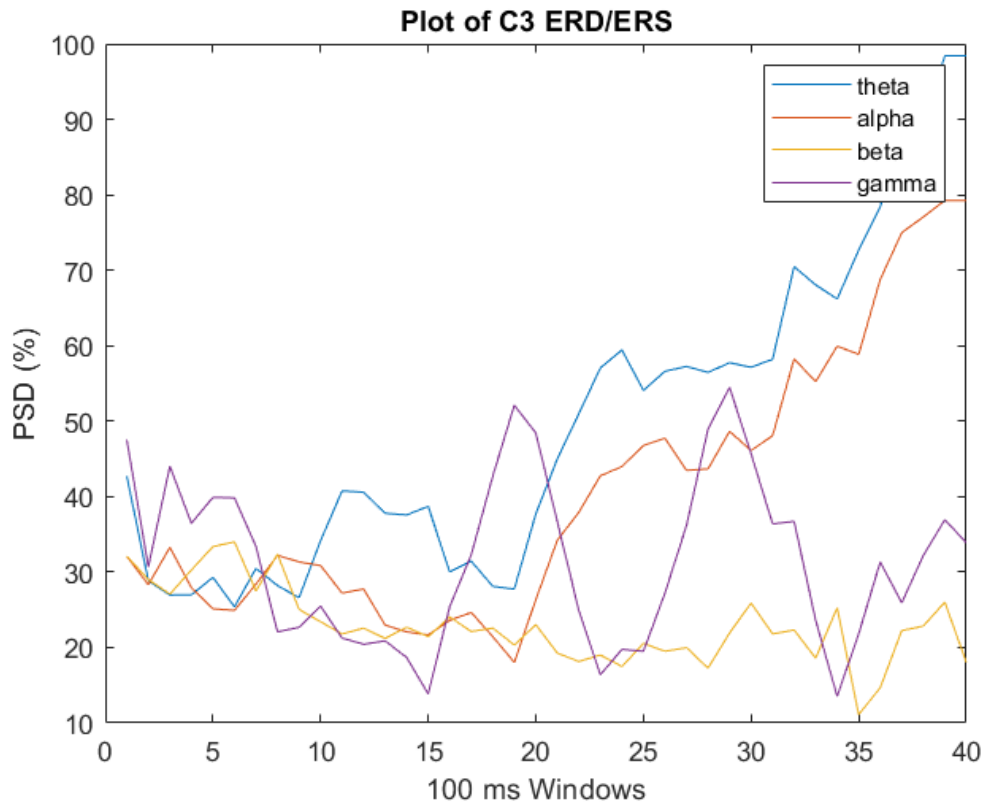


Fig 3.4: ERD/ERS Plot of C3

It can be observed that both alpha and beta exhibit synchronization during the Rest period (0 – 2s) and exhibit desynchronization during the active period (2 – 4s). We can now examine ways of extracting meaningful information from the ERD/ERS signals from the alpha and beta bands.

3.3.3 Classification Algorithms

One of the widely used and effective classification algorithms is the kNN. We perform a 10-fold cross validation using the dataset. In cross-validation, every fold will train the model using the in-fold observations and perform a prediction using the observations from the out-fold.

- **Power Band Features**

The kNN model was first trained using the PSD values of the two states. Figure 3.5 shows the Scatter Plot and the Confusion Matrix

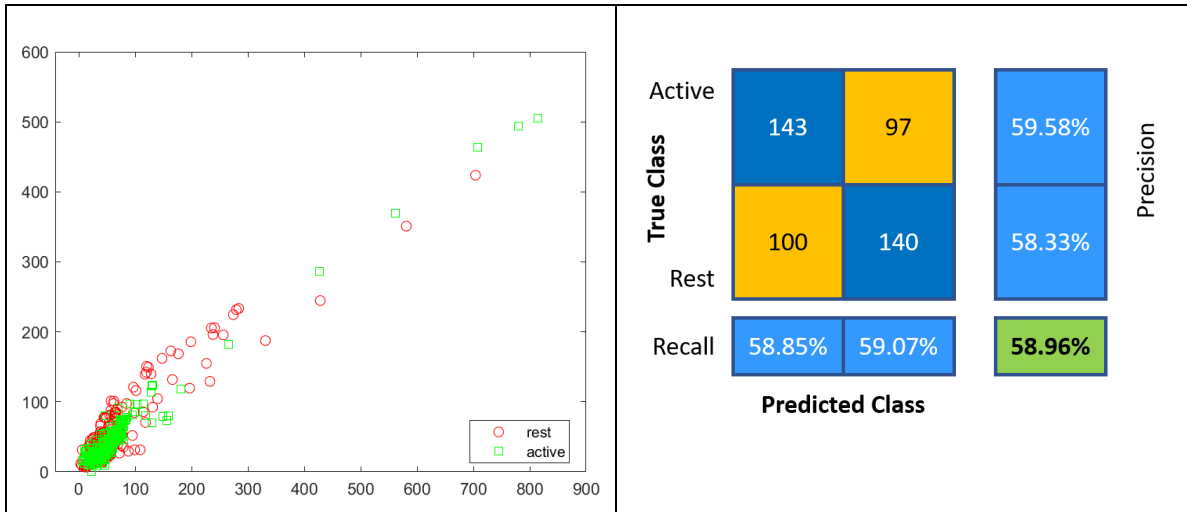


Fig 3.5: Scatter Plot and Confusion Matrix for kNN Classification using Alpha and Beta PSD features

The kfoldLoss (L) which indicates the average loss across all the folds was 0.4104. Using the Confusion Matrix, we can calculate the classification accuracy using the formula below.

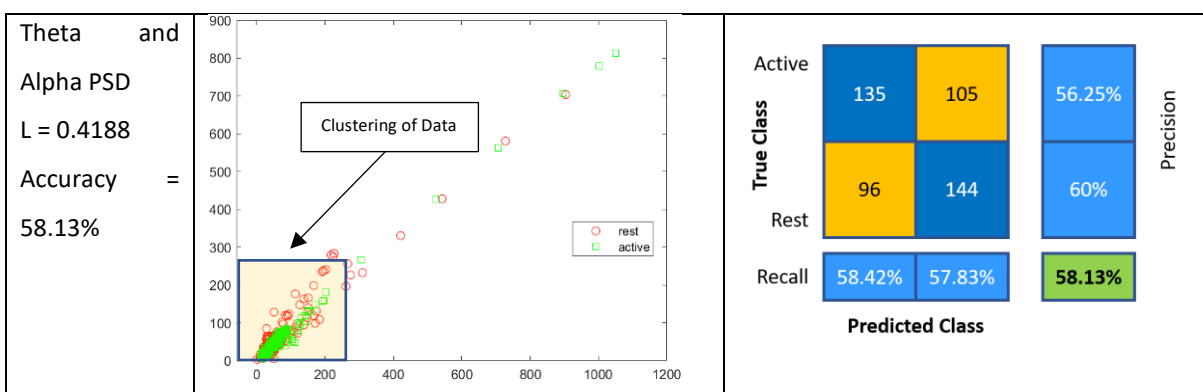
$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad \text{Eq. 3.4}$$

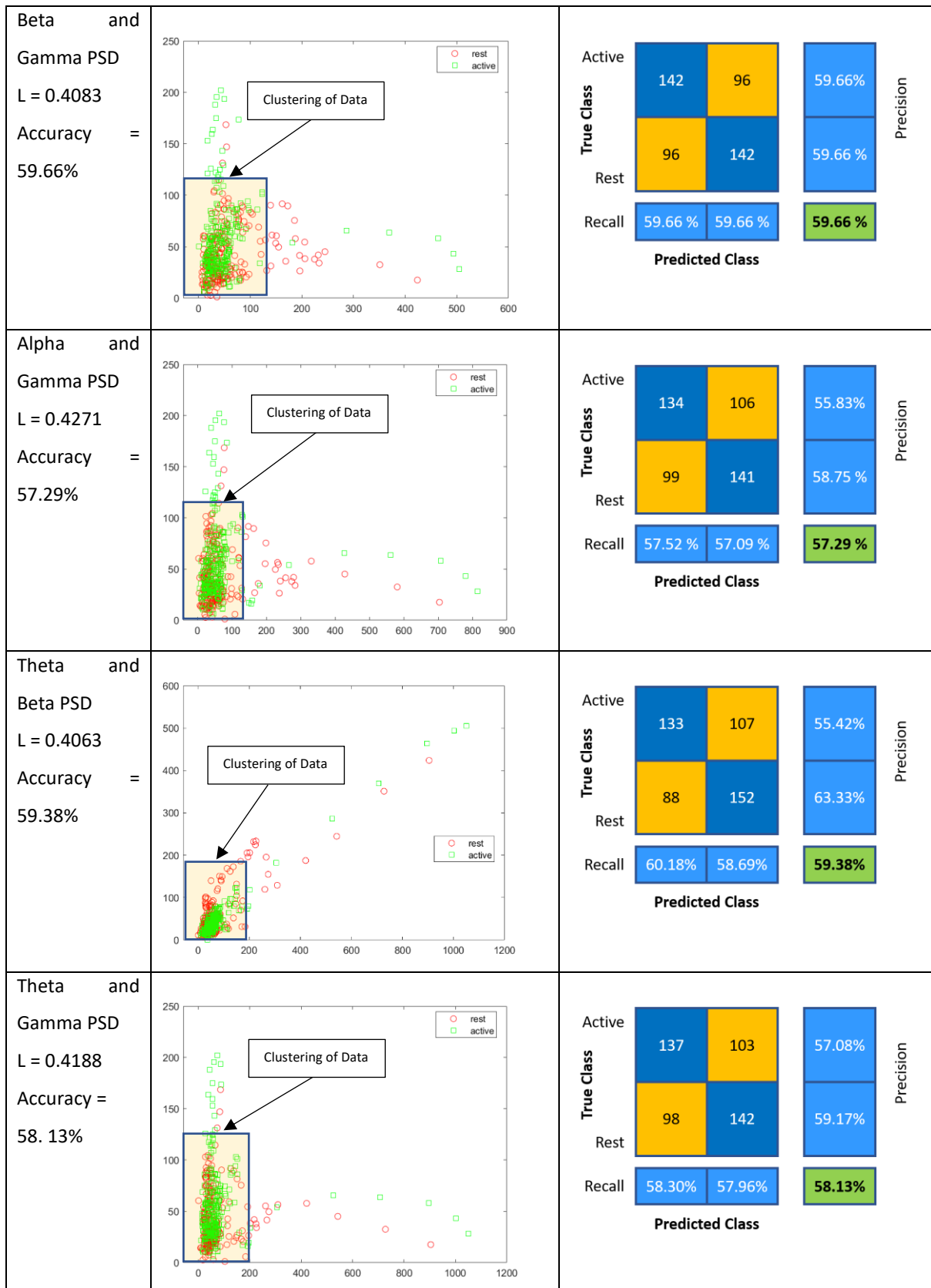
Where TP = True Positive, TN = True Negative, FP = False Positive and FN = False Negative

The accuracy for the above classification is $\frac{(143+140)}{(143+140+97+100)} = \frac{283}{480} = 58.96\%$

From the Scatterplot, we can observe that there is significant overlap between the datapoints for both states, hence the poor accuracy. Further analysis was done using different combination of PSD features. Table 3.2 shows the results.

Table 3.2: Plots and Values for various PSD Combinations



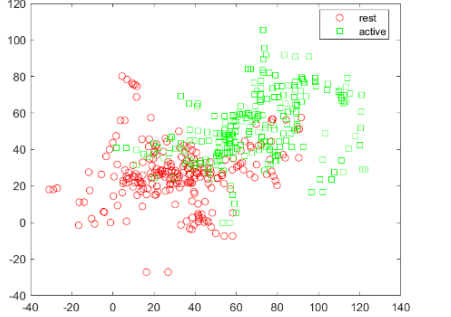
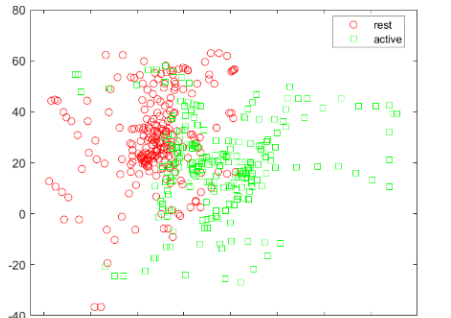
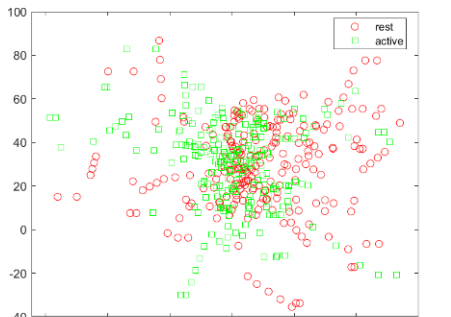
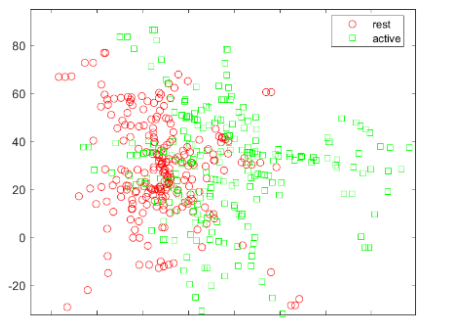


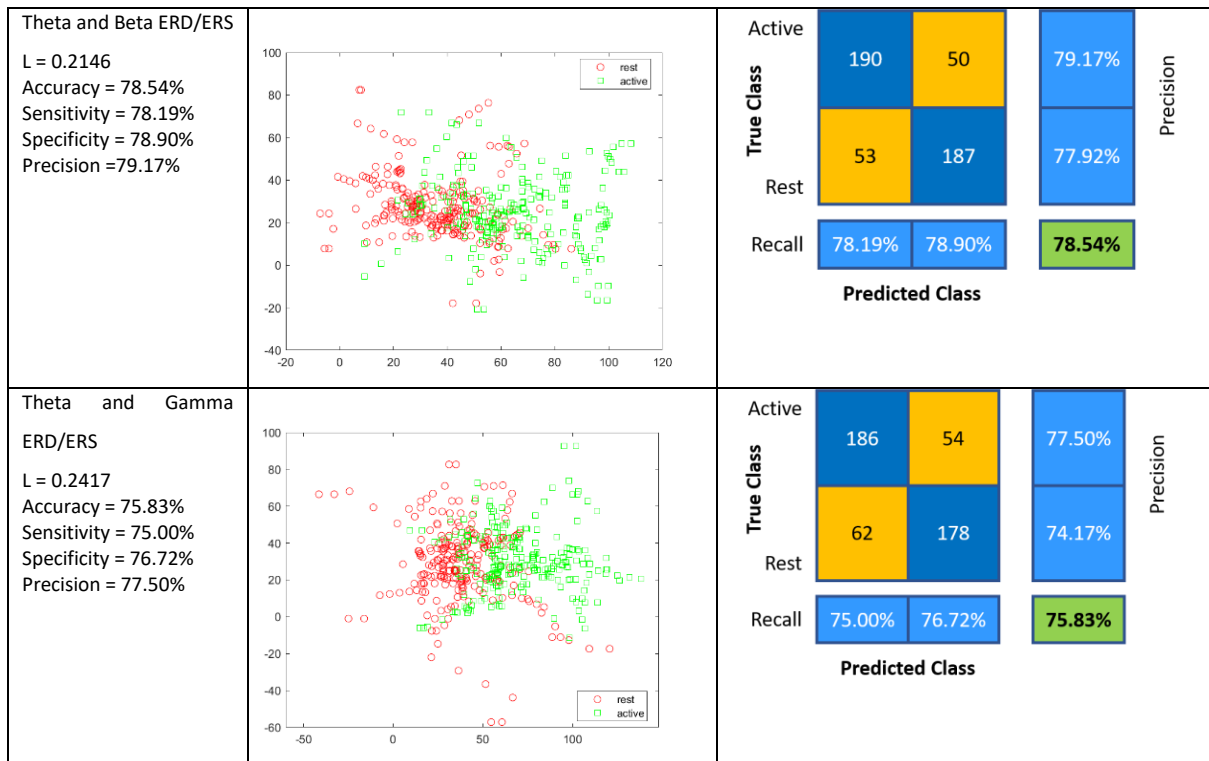
Accuracy Range = 57.29% - 59.17%

- **ERD/ERS Features**

The kNN model is trained using ERD/ERS features and a similar classification process is carried out. Table 3.3 shows the results obtained for the various Power Bands.

Table 3.3: Plots and Values for various ERD/ERS Combinations

<p>Theta and Alpha ERD/ERS</p> <p>L = 0.1771 Accuracy = 82.29% Sensitivity = 82.16% Specificity = 82.43% Precision = 82.5%</p>		<table border="1"> <tr> <td rowspan="2">True Class</td> <td>Active</td> <td>198</td> <td>42</td> <td>82.5%</td> <td rowspan="2">Precision</td> </tr> <tr> <td>Rest</td> <td>43</td> <td>197</td> <td>82.08%</td> </tr> <tr> <td>Recall</td> <td></td> <td>82.16%</td> <td>82.43%</td> <td>82.29%</td> <td></td> </tr> <tr> <td colspan="6" style="text-align: center;">Predicted Class</td> </tr> </table>	True Class	Active	198	42	82.5%	Precision	Rest	43	197	82.08%	Recall		82.16%	82.43%	82.29%		Predicted Class					
True Class	Active	198		42	82.5%	Precision																		
	Rest	43	197	82.08%																				
Recall		82.16%	82.43%	82.29%																				
Predicted Class																								
<p>Alpha and Beta ERD/ERS</p> <p>L = 0.1729 Accuracy = 82.71% Sensitivity = 83.12% Specificity = 82.31% Precision = 82.08%</p>		<table border="1"> <tr> <td rowspan="2">True Class</td> <td>Active</td> <td>197</td> <td>43</td> <td>82.08%</td> <td rowspan="2">Precision</td> </tr> <tr> <td>Rest</td> <td>40</td> <td>200</td> <td>83.33%</td> </tr> <tr> <td>Recall</td> <td></td> <td>83.12%</td> <td>82.31%</td> <td>82.71%</td> <td></td> </tr> <tr> <td colspan="6" style="text-align: center;">Predicted Class</td> </tr> </table>	True Class	Active	197	43	82.08%	Precision	Rest	40	200	83.33%	Recall		83.12%	82.31%	82.71%		Predicted Class					
True Class	Active	197		43	82.08%	Precision																		
	Rest	40	200	83.33%																				
Recall		83.12%	82.31%	82.71%																				
Predicted Class																								
<p>Beta and Gamma ERD/ERS</p> <p>L = 0.3146 Accuracy = 68.54% Sensitivity = 67.45% Specificity = 69.78% Precision = 71.67%</p>		<table border="1"> <tr> <td rowspan="2">True Class</td> <td>Active</td> <td>172</td> <td>68</td> <td>71.67%</td> <td rowspan="2">Precision</td> </tr> <tr> <td>Rest</td> <td>83</td> <td>157</td> <td>65.42%</td> </tr> <tr> <td>Recall</td> <td></td> <td>67.45%</td> <td>69.78%</td> <td>68.54%</td> <td></td> </tr> <tr> <td colspan="6" style="text-align: center;">Predicted Class</td> </tr> </table>	True Class	Active	172	68	71.67%	Precision	Rest	83	157	65.42%	Recall		67.45%	69.78%	68.54%		Predicted Class					
True Class	Active	172		68	71.67%	Precision																		
	Rest	83	157	65.42%																				
Recall		67.45%	69.78%	68.54%																				
Predicted Class																								
<p>Alpha and Gamma ERD/ERS</p> <p>L = 0.2708 Accuracy = 72.92% Sensitivity = 73.50% Specificity = 72.36% Precision = 71.67%</p>		<table border="1"> <tr> <td rowspan="2">True Class</td> <td>Active</td> <td>172</td> <td>68</td> <td>71.67%</td> <td rowspan="2">Precision</td> </tr> <tr> <td>Rest</td> <td>62</td> <td>178</td> <td>74.17%</td> </tr> <tr> <td>Recall</td> <td></td> <td>73.50%</td> <td>72.36%</td> <td>72.92%</td> <td></td> </tr> <tr> <td colspan="6" style="text-align: center;">Predicted Class</td> </tr> </table>	True Class	Active	172	68	71.67%	Precision	Rest	62	178	74.17%	Recall		73.50%	72.36%	72.92%		Predicted Class					
True Class	Active	172		68	71.67%	Precision																		
	Rest	62	178	74.17%																				
Recall		73.50%	72.36%	72.92%																				
Predicted Class																								



Accuracy Range = 68.54% - 82.71%

- **Detection of Gamma Peak**

From Figure 3.6, it can be observed that there is a Gamma Peak around the time where the Active state occurs.

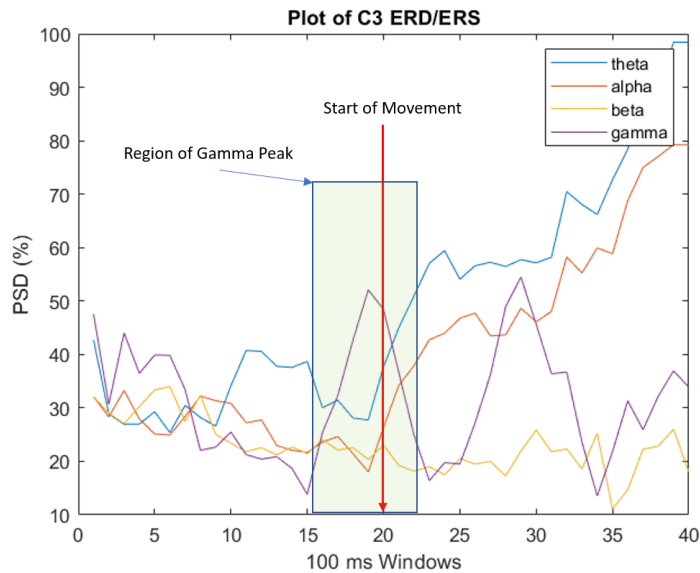
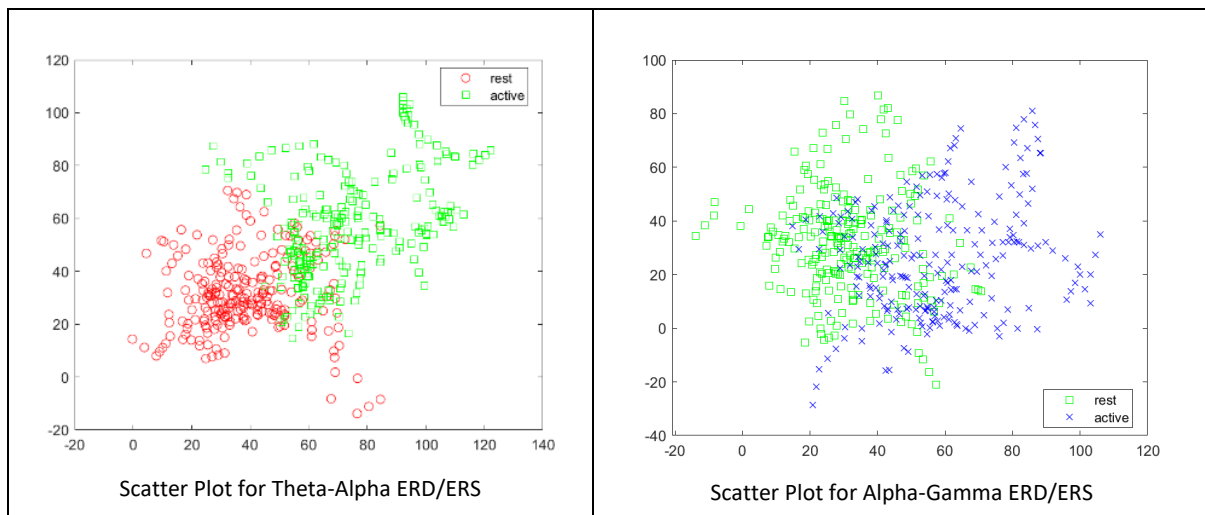


Fig 3.6: Gamma Peak Observation around Rest/Active Transition

Integrating the Gamma ERD/ERS values for the classification yields the following results for the following two combinations that showed the best results so far.

Table 3.4: Results for Theta-Alpha-Gamma ERD/ERS features



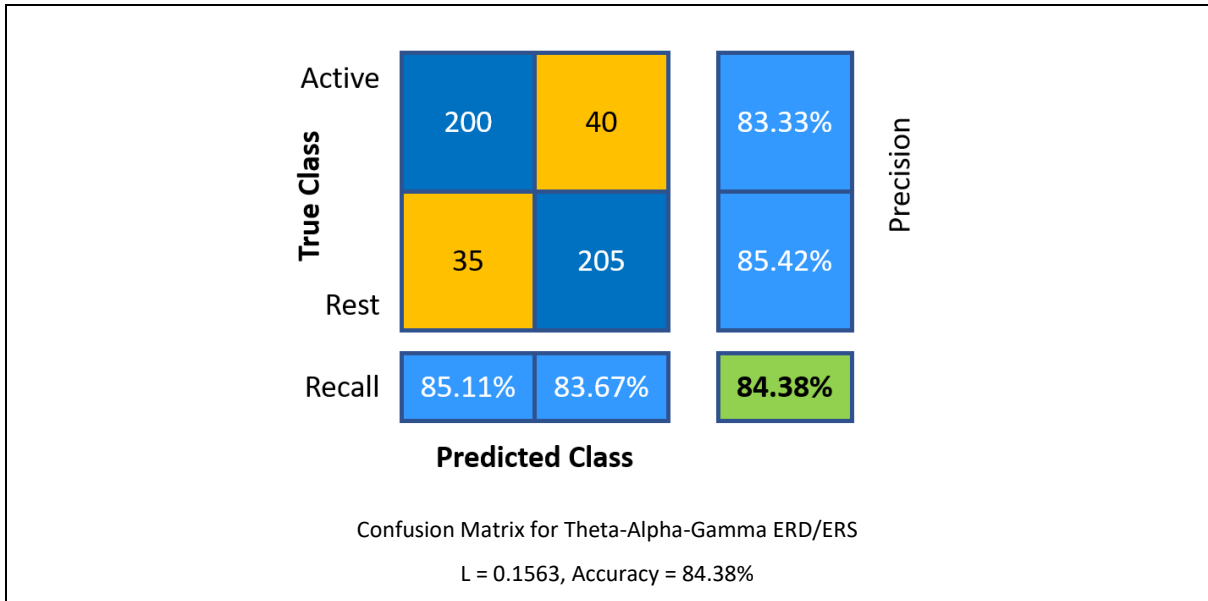
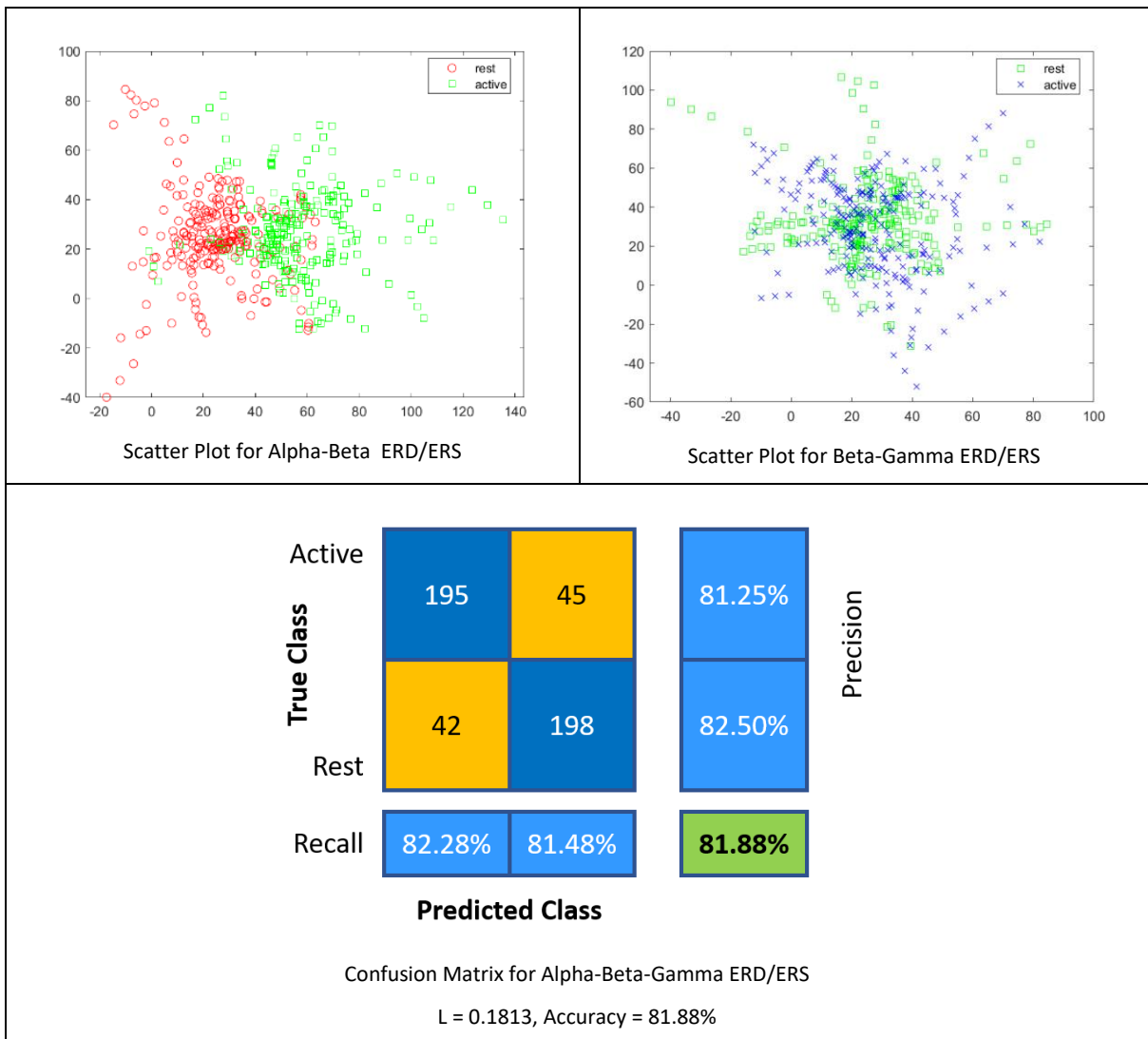


Table 3.5: Results for Alpha-Beta-Gamma ERD/ERS features



It can be observed that the results shown in Table 3.4 for the Theta-Alpha-Gamma ERD/ERS and in Table 3.5 for the Alpha-Beta-Gamma ERD/ERS are both comparable to the earlier results obtained in Table 3.3 where the Gamma PSD features were not included. This is because the Gamma signal is not entirely correlated with the Alpha and Beta bands. The peak of the Gamma is the key feature to be captured to be an indicator for the Active State. It was also observed that the peak occurs within a range of $\pm 100\text{ms}$ from the moment of the Active state.

3.3.4 Novel Signal Processing and Classification Model

Based on the results gathered, there is a need to detect the Gamma values and correlate its peak with the ERD/ERS events happening between the other power bands. The proposed model, as shown in Figure 3.7, provides a 50ms delay for each ' z^{-1} ' unit. This allows us to capture the windowed signal and perform a slope and peak detection. The output data from the block is then fed to the kNN classifier together with the other Power Bands.

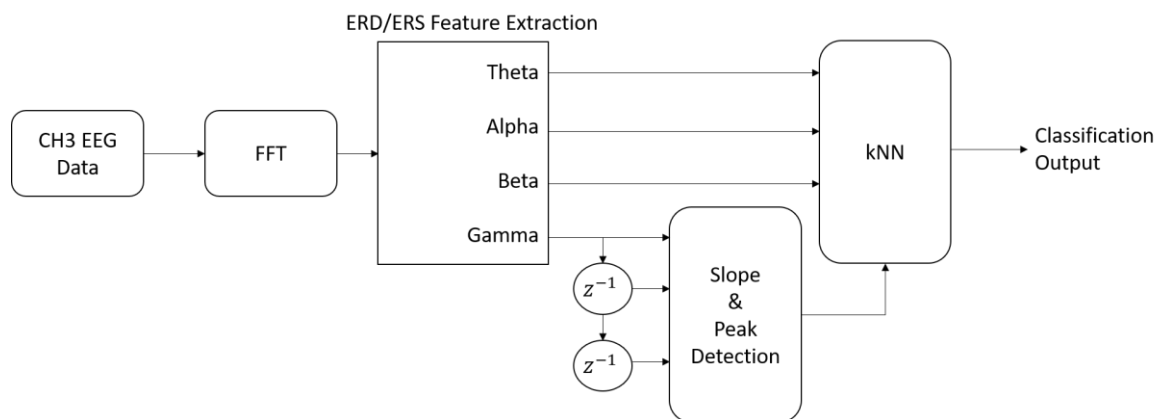


Fig 3.7: ERD/ERS with Gamma Peak Classification Model

- **Slope and Peak Detection**

The slope and peak detection are performed using the following steps for a vector X with length m .

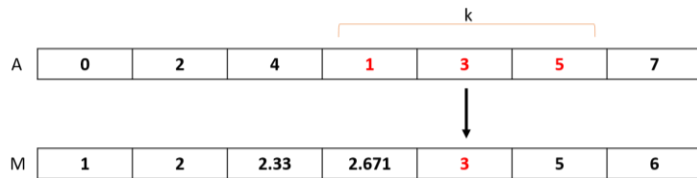
Step 1: Diff Computation

$$Diff(Y) = [X(2) - X(1) \quad X(3) - X(2) \quad \dots \quad X(m) - X(m - 1)] \quad \text{Eq. 3.5}$$

Where $X(m)$ refers to the m_{th} sample

Step 2: Moving Average

Using a window length of 3, the average is computed using the centre value its surrounding neighbours using the following approach.



$$\mu = \frac{1}{N} \sum_{i=1}^N A_i \quad \text{Eq. 3.6}$$

Where μ is the Moving Average, N is the number of samples and A_i is each sample

Step 3: Peak Detection

Peak Detection is then carried out by checking each data point across its neighbours. The threshold value selected is the median value of the last 5 samples. If the current peak value is greater than 30% of the threshold value, it will be detected as a valid peak. Figure 3.8 shows an example of the peaks being detected.

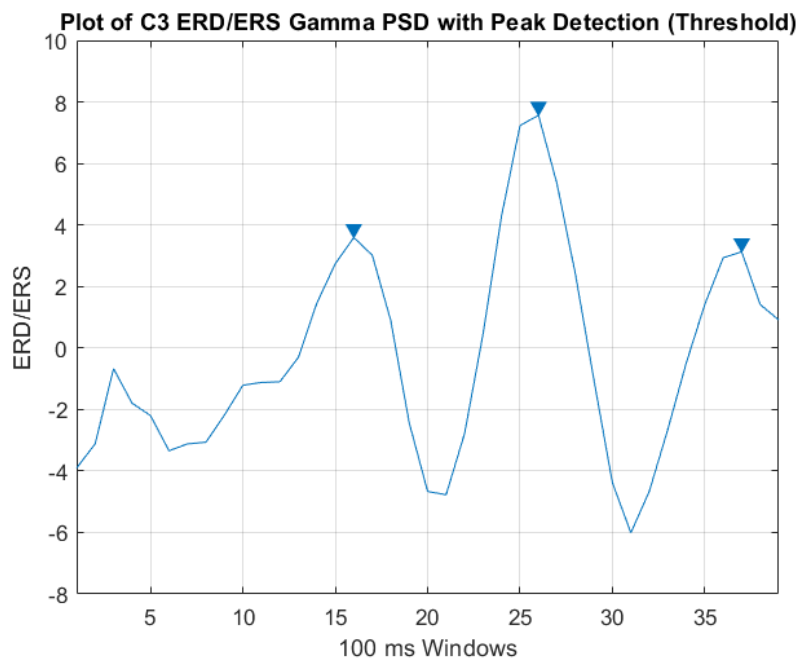


Fig 3.8: Plot of C3 ERD/ERS Gamma PSD with Peak Detection Threshold of 1

3.4 Results and Discussion

The results of the improved ERD/ERS model with Gamma Peak are as shown below in Table 3.6.

Table 3.6: Results for ERD/ERS with Gamma Peak

<p>Theta-Alpha-Gamma ERD/ERS</p> <p>L = 0.1042</p> <p>Accuracy = 89.58%</p> <p>Sensitivity = 89.26%</p> <p>Specificity = 89.92%</p> <p>Precision = 90.00%</p>	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td rowspan="2" style="writing-mode: vertical-rl; transform: rotate(180deg);">True Class</td> <td>Active</td> <td style="background-color: #0070C0; color: white; text-align: center;">216</td> <td style="background-color: #FFC000; color: white; text-align: center;">24</td> <td rowspan="2" style="writing-mode: vertical-rl; transform: rotate(180deg);">Precision</td> </tr> <tr> <td>Rest</td> <td style="background-color: #FFC000; color: white; text-align: center;">26</td> <td style="background-color: #0070C0; color: white; text-align: center;">214</td> </tr> <tr> <td>Recall</td> <td style="background-color: #0070C0; color: white; text-align: center;">89.26%</td> <td style="background-color: #0070C0; color: white; text-align: center;">89.92%</td> <td style="background-color: #70AD47; color: white; text-align: center;">89.58%</td> <td></td> </tr> <tr> <td></td> <td colspan="2" style="text-align: center;">Predicted Class</td> <td></td> <td></td> </tr> </table>	True Class	Active	216	24	Precision	Rest	26	214	Recall	89.26%	89.92%	89.58%			Predicted Class			
True Class	Active		216	24	Precision														
	Rest	26	214																
Recall	89.26%	89.92%	89.58%																
	Predicted Class																		
<p>Alpha-Beta-Gamma ERD/ERS</p> <p>L = 0.0771</p> <p>Accuracy = 92.29%</p> <p>Sensitivity = 92.47%</p> <p>Specificity = 92.17%</p> <p>Precision = 92.08%</p>	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td rowspan="2" style="writing-mode: vertical-rl; transform: rotate(180deg);">True Class</td> <td>Active</td> <td style="background-color: #0070C0; color: white; text-align: center;">221</td> <td style="background-color: #FFC000; color: white; text-align: center;">19</td> <td rowspan="2" style="writing-mode: vertical-rl; transform: rotate(180deg);">Precision</td> </tr> <tr> <td>Rest</td> <td style="background-color: #FFC000; color: white; text-align: center;">18</td> <td style="background-color: #0070C0; color: white; text-align: center;">222</td> </tr> <tr> <td>Recall</td> <td style="background-color: #0070C0; color: white; text-align: center;">92.47%</td> <td style="background-color: #0070C0; color: white; text-align: center;">92.17%</td> <td style="background-color: #70AD47; color: white; text-align: center;">92.29%</td> <td></td> </tr> <tr> <td></td> <td colspan="2" style="text-align: center;">Predicted Class</td> <td></td> <td></td> </tr> </table>	True Class	Active	221	19	Precision	Rest	18	222	Recall	92.47%	92.17%	92.29%			Predicted Class			
True Class	Active		221	19	Precision														
	Rest	18	222																
Recall	92.47%	92.17%	92.29%																
	Predicted Class																		

It can be observed that the overall accuracy is now in the range of 89.58% to 92.29%. This is an increase of approximately 7.7% - 7.91%.

It can be deduced that relying solely on PSD parameters for motor movement classification is not practical from the results obtained. The ERD/ERS feature set can provide us with a substantial improvement in the overall results. The Gamma peak around the region of active movement is an effective indicator of the user's intention. The proposed model with slope

and peak detection, combined with temporal shift, can better capture this peak and use it as an essential feature in our classifier. A comparative study of the results was performed with research papers with comparable results. For datasets that are publicly available, the methodology proposed in this paper was used to obtain results to provide a fair comparison. The results of this comparative study are summarized in Table 3.7

In [191], the temporal correlation of EEG signals was explored across two different datasets. The private dataset was based on finger-tapping. The public dataset [192] provides a rest state and active state that comprises four other tasks. The proposed method generated a classification accuracy of 81.24% for the Rest and Active states as compared to the accuracy of 72.24% - 76.07%. In [193], 25 participants performed 11 different movement tasks and multiple physiological signals were captured during the experiments. The classification accuracy for the 'Active' State task was within the range of 58% - 59%, with a S.D. of around 0.04. The proposed method generated a classification accuracy of 77.28%. From Table 3.7, we can observe that the proposed method has achieved a higher classification accuracy with a single channel.

Table 3.7: Comparison of Results

Ref	Metrics	No. of Channels	Dataset	Features	Classifier
[191]	Accuracy = 72.24% - 73 %	3	[192]	LRTC	LDA
[191]	Accuracy = 75.69% - 76.07 %	3	Private	LRTC	LDA
[193]	Accuracy = 58% - 59%	60	[193]	CSP	LDA
*Current Paper	Accuracy = 92.99% Sensitivity = 92.47% Specificity = 92.12% Precision = 92.08%	1	[188]	ERD/ERS	kNN

*Current Paper	Accuracy = 81.24%	1	[192]	ERD/ERS	kNN
	Sensitivity = 80.12%				
	Specificity = 79.41%				
	Precision = 80.15%				
*Current Paper	Accuracy = 77.28%	1	[193]	ERD/ERS	kNN
	Sensitivity = 75.51%				
	Specificity = 76.43%				
	Precision = 76.89%				

Discussion

It can be seen that the proposed Gamma-Peak methodology is able to generate a feature set that is more clearly separable for the various classes. Using a single channel, we are able to achieve high-levels of accuracy and demonstrate its consistency across different datasets. We will now study methods of decoding and classifying EMG signals.

Publication:

A Novel Event-Related Desynchronization/Synchronization with Gamma Peak EEG model for Motor State Identification

2021 International Conference on Computational Science and Computational Intelligence (CSCI)

2021-12 | Conference paper

DOI: [10.1109/csci54926.2021.00245](https://doi.org/10.1109/csci54926.2021.00245)

CONTRIBUTORS: Ravi Suppiah; Ravi Suppiah; Anurag Sharma; Noori Kim; Khalid Abidi

Chapter 4

EMG Decoding and Classification

Movement of any part of the body is representative of neuromuscular activities. Capturing such signals is critical to designing systems that act upon the intent of the user. They also play a key role in the rehabilitation assessment process and can help to provide a more objective assessment of a patient's recovery. The use of EEG signals to detect a person's intention to provide rehabilitation was explored in the earlier chapter. In this chapter we will explore another approach with the use of EMG signals.

4.1 Physical Movement

The human skeletal muscular framework was first modelled by in 1938 and was known as the Hill-type model [93]. Three main electrical elements of series, parallel and contractile were used to model and replicate the nature of human muscles. This model was used as the basis for the design and implementation of a exoskeleton arm in [94], and was used in subsequent research activities. However, the complexity of the several arm parameters made it challenging as they were very dependent on an individual's physiological characteristics.

When using EMG signals for rehabilitation, we are in understanding the neuromuscular activation that is observed in the various muscles during different activities, such as, functional movements, activities of daily living, and training. There are two main classes of resistance-training exercises, and they are isometric and isotonic contractions. In the field of

rehabilitation, both types of contractions are widely used, while isotonic contractions are commonly observed in areas where the focus is on athletic and strength goals.

4.1.1 Isotonic

In isotonic contractions, we will observe muscular contractions that oppose resistance, and the length of the muscle will also change. By varying the muscle's length, the contraction is able to generate a force. The type of contraction can be eccentric or concentric. In concentric contractions, the muscles tend to shorten [95]. During this time, the tension within the muscle is constant [96]. In eccentric contraction, the muscles elongate as they face resistance. When the external force is greater than what can be generated by the muscle, it goes through a forced lengthening. Many normal activities, like walking, are considered eccentric and are thus a popular area of study [97]. Many muscular injuries are also associated with eccentric contractions [98].

4.1.2 Isometric

In isometric contractions, the muscle length does not change, however, the energy and tension within the muscles keep fluctuating, and this allows the production of a force. Typically, isometric contractions are observed when there is an action performed towards a fixed object without any resultant movement.

4.1.3 Application of Isotonic and Isometric Contractions

The study of muscular fatigue and neuromuscular diseases can be performed through EMG signals that are obtained through isotonic conditions [99, 100]. Many assistive robotic systems, exoskeletons, and lower-limb orthoses, are designed based on isotonic and isometric contractions [101, 102]. A wide variety of techniques have been described in various literature that focus on applying these signals for the purpose of controlling a rehabilitative robotic device [103-106].

4.1.4 Electromyography (EMG)

The EMG is a complex signal [107] and has many dependencies on the physiological and anatomical properties of the underlying muscles. The most convenient approach to obtaining EMG signals is through the surface of the skin. Thus, it is also known as sEMG (surface EMG). However, the signals obtained through this method also tend to have a lot of noise due to

inputs from other neighbouring motor units. A wide variety of signal processing and feature extraction techniques have to be utilized to obtain a detailed analysis of the data obtained.

In our brains, the nerves conduct electrical potentials to convey information. A similar phenomenon is observed with muscles. The motor unit action potential (MUAP) is the combined effect of the various muscle fibres associated with a single motor unit. It can be modelled using this formula:

$$x(n) = \sum_{r=h(r)0}^{N-1} h(r)e(n-r) + w(n) \quad \text{Eq. 4.1}$$

In the formula, N is the number of motor units that are firing, $w(n)$ represents the white noise, $h(r)$ represents MUAP, $e(n)$ is the processed point and $x(n)$ is the modelled EMG signal. Generally, many integrated sensors may perform some basic signal processing techniques using some HW. These include amplification, filtering to remove artefacts and amplitude detection. They may also provide the raw data to allow custom signal processing and feature extraction techniques.

- **Signal Overview**

When muscle fibres contract, an electric potential is generated within them, and this is what the EMG electrode picks up. EMG sensors are categorized into imEMG (intramuscular EMG) and sEMG (surface EMG). Though imEMG sensors can provide signals with better SNR [194], their setup is difficult and causes a great deal of discomfort to the user. As a result, the use of sEMG signals is a preferred choice for many such applications. The electrodes generally come in a pair of poles, though they also have monopole types. The EMG signal generated is greatly impacted by the distance between poles and their diameters [195].

Though the intention of using these sensors is to explore the electric potentials within the nerves, the sEMG electrodes are actually measuring the muscle cell potentials when the motor nerves excite them. As there is a high correlation between these two readings [196], sEMG provides a practical approach to decoding muscle activity data.

The various muscle groups in the forearm are shown in Figure 4.1. Electrodes can be placed in a targeted approach where it is positioned exactly above the specific muscle we are interested in. It can also be placed in a non-targeted fashion where the location is in the vicinity of the specific muscle. It was demonstrated that it is possible for both techniques to generate good results if appropriate signal processing and pattern recognition techniques are used.

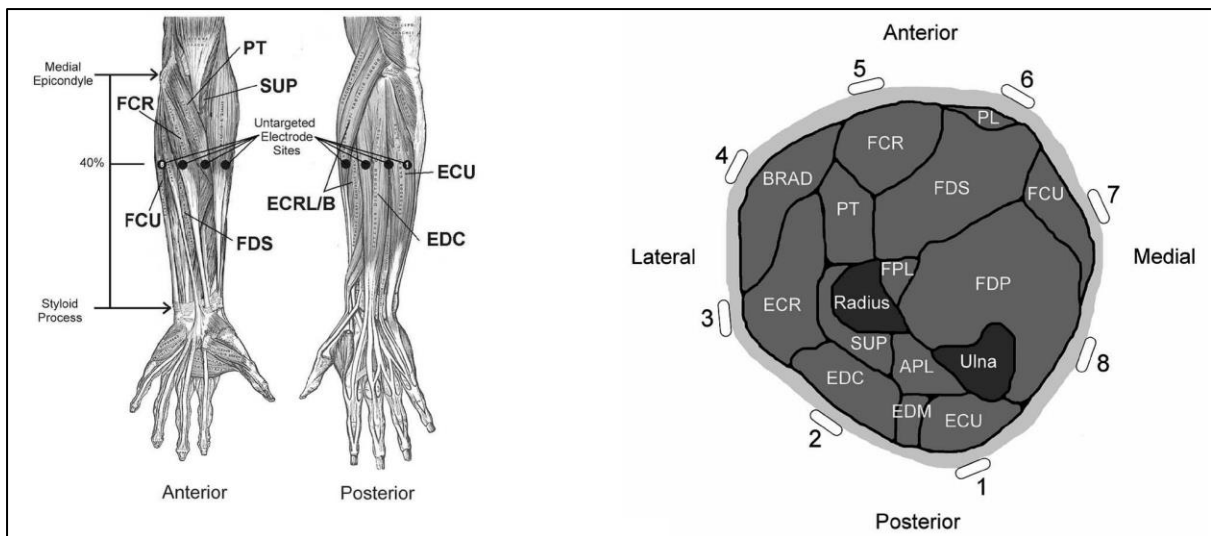


Figure 4.1: Longitudinal and transverse representation of the forearm muscles. [194]

- **Signal Processing**

Many different approaches have been described by authors in this area, and it is highly dependent on the type of sensor used and its related data acquisition hardware. The common techniques used are amplification, band-pass filtering and power-line interference suppression, and digitization. In amplification, an operational amplifier can be used to provide the required gain [197]. As the actual voltages are very small, the amplification can go be in the range of a few hundred to a few thousand. The cut-off frequencies, for the band-pass filtering can be in the range of 10 to 50Hz for the lower cut-off and around 400 to 500 Hz for the upper cut-off [198-200]. The power line interference must also be suppressed. In most papers, the authors have chosen to use the notch-filter to easily perform his task. Other techniques that have been explored in this area include Regression [201], Spectrum

Interpolation [202] and Adaptive Laguerre filters [203]. The filtered signal is then digitized for further processing. The sampling rate is generally in the 1kHz to 2kHz range.

- **Sensor Arrays**

As an alternative to focused electrode placement, sensor arrays allow us to capture signals from a range of sensors. This allows us to capture the desired EMG signals from multiple sources. Such an approach adds computational complexity to the overall system as well as the cost of any practical solution. In [204], the authors used 57 EMG channels. They were able to demonstrate a high degree of classification accuracy for the classification of gestures. However, continuous retraining was required for each session. High-density electrode arrays were studied in [205-207]. These techniques provided a high degree of accuracy with a corresponding increase in the post-processing computational power requirements. The authors suggested using feature reduction techniques to reduce the number of EMG channels.

4. 2 Dataset

The dataset selected for this research is obtained from Luciw, M et al. [188], and is as described in Section 3.2

The 5 EMG sensors were placed on pertinent right arm muscles. The chosen locations were the anterior first dorsal interosseus, common extensor digitorum, flexor digitorum, brachioradial and the deltoid muscles. The positions of the sensor on a participant can be seen in Figure 4.2.

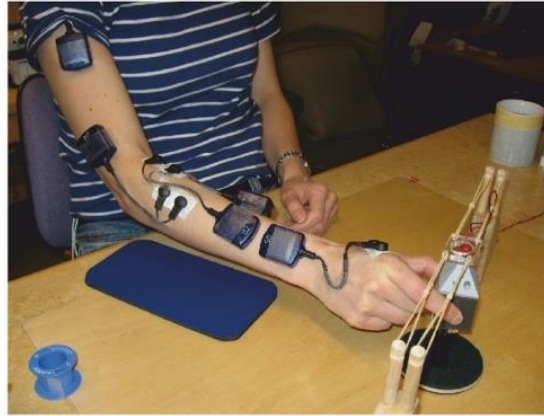


Fig 4.2: EMG sensor placement: 1-anterior deltoid, 2-brachioradialis, 3-flexor digitorum, 4-common extensor digitorum, 5-first dorsal interosseus

In each trial, the participants performed the same task of moving their arm forward to pick-up an object to a fixed height. This was cued by an LED being turned ON. When the LED is turned OFF, the participant would lower the object and return the arm back to the resting state. The sequence of the movements is shown in Fig 4.3.

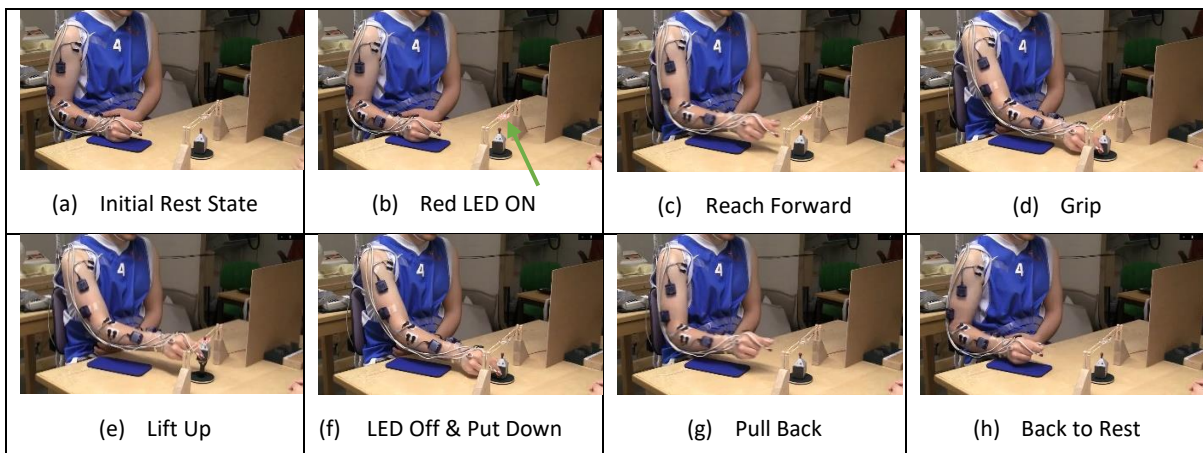


Figure 4.3: Sequence of ARM Movements

The raw data captured from the EMG sensors is shown in Figure 4.4. The green line indicates the time where the LED was turned ON, and the red line indicates the time when the LED was turned OFF.

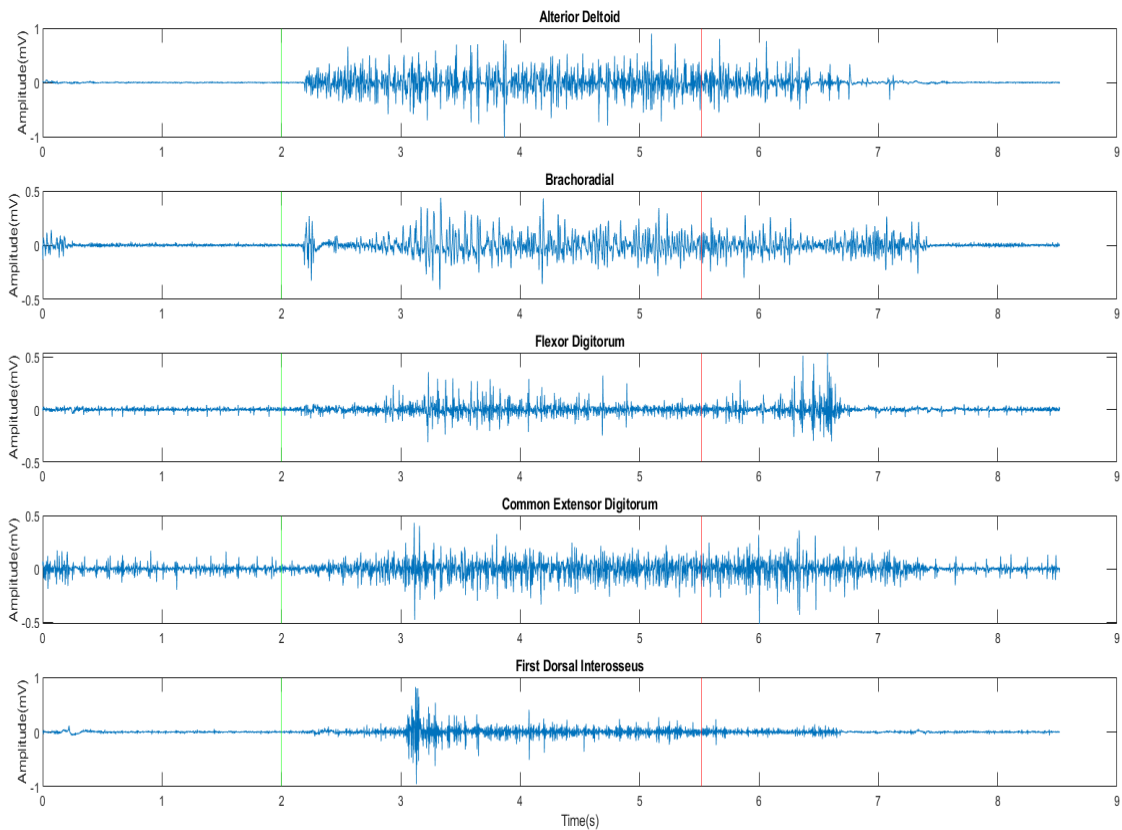


Fig 4.4: Raw EMG Channels

4. 3 Signal Pre-Processing

4.3.1 Band-Pass Filtering

The raw data is pre-processed to aid in the feature extraction stage. In the first step of pre-processing, a 4th order Butterworth Bandpass filter with cut-frequencies of 10Hz and 500Hz is used. After filtering, the signal still has both positive and negative values. In order to remove the negative portion, the signal is rectified by taking the absolute value of the signal.

4.3.2 RMS Value

The EMG time-domain signal is “noisy” with many spikes and troughs at high frequency. In order to get a better understanding of the signal’s response, the Root-Mean-Square of the Signal is computed. This gives a smoothed response of the data while preserving the important transients. The waveforms for the various stages (Anterior Deltoid) are shown in Fig 4.5.

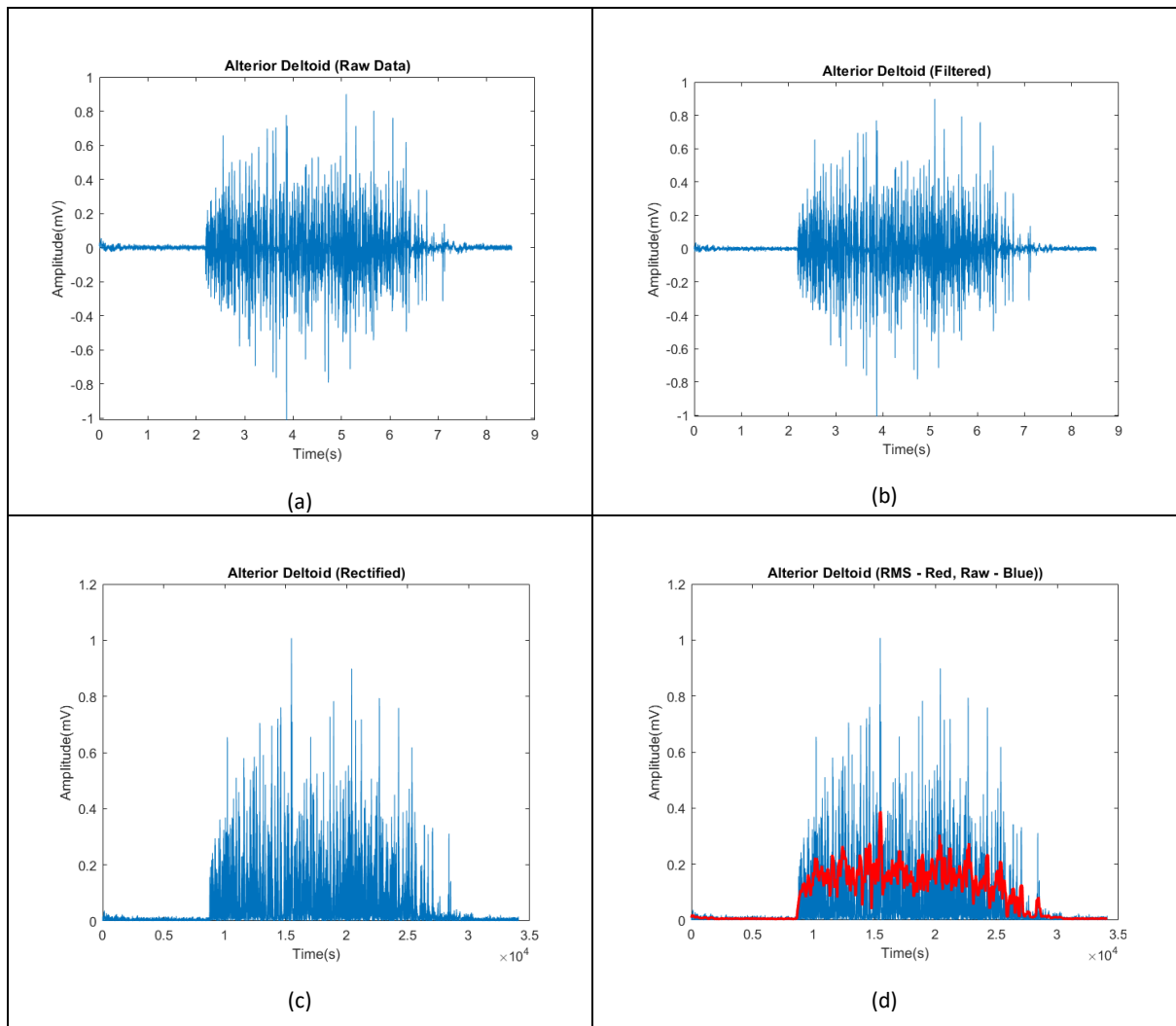


Fig 4.5: (a) Raw Data, (b) Filtered Data (c) Rectified Data (d) RMS Data

The complete flow of the signal processing blocks is shown in Figure 4.6

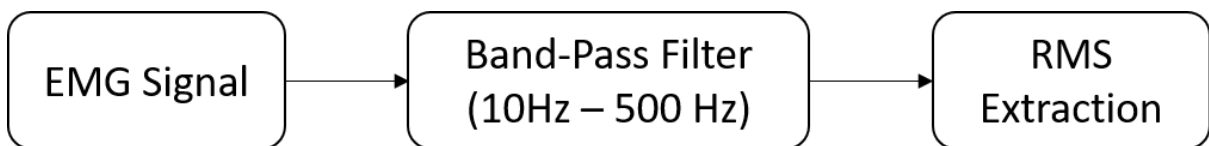


Figure 4.6: Signal Pre-Processing Blocks

The original signals with their RMS signals for all the EMG sensors are shown in Fig 4.7

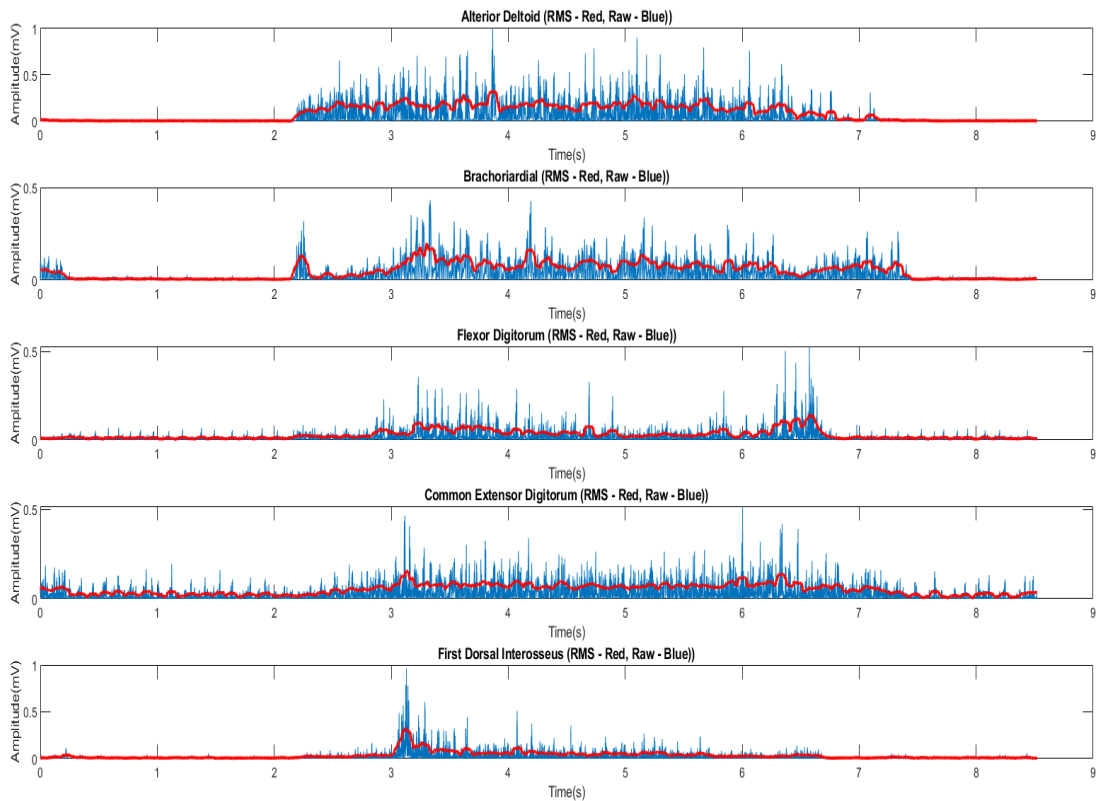


Fig 4.7: All EMG Channels – RMS – Red, Raw – Blue

4. 4 Feature Extraction

4.4.1 Discrete Wavelet Transform

For any given signal s that has a length of N , the Discrete Wavelet Transform (DWT) will be made up of $\log_2 N$ steps. We start with the original signal s , and the first step generate two sets of coefficients, the Detail Coefficients cD_i and the Approximation Coefficients cA_i . This is obtained by performing a convolution with the high-pass filter HiD , and the low-pass filter LoD . The results of the filtering operation are then down-sampled to obtain cD_i and cA_i . This is shown in Figure 4.8.

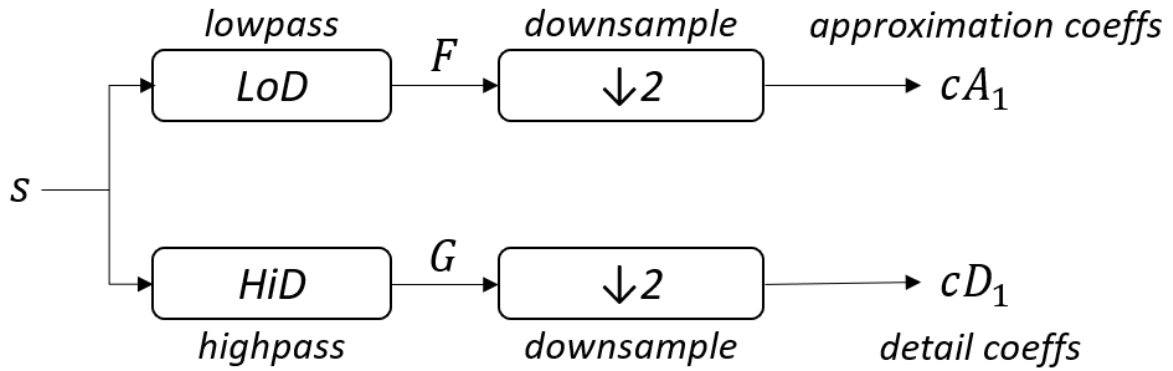


Figure 4.8: DWT Operation

A four-level DWT decomposition is performed on the signal and a decomposition vector containing the approximation and detailed coefficients is obtained as shown in Figure 4.9.

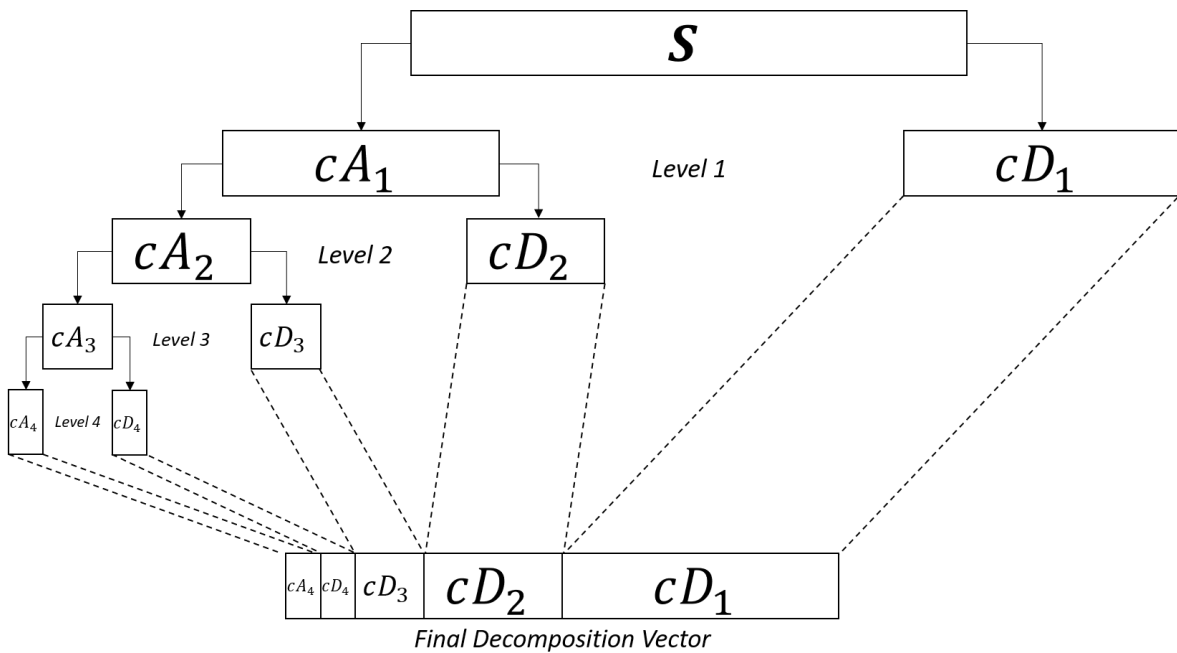


Figure 4.9: Four-Level DWT Decomposition

In Figure 4.10, we can see the wavelet transformed signals for the AD channel.

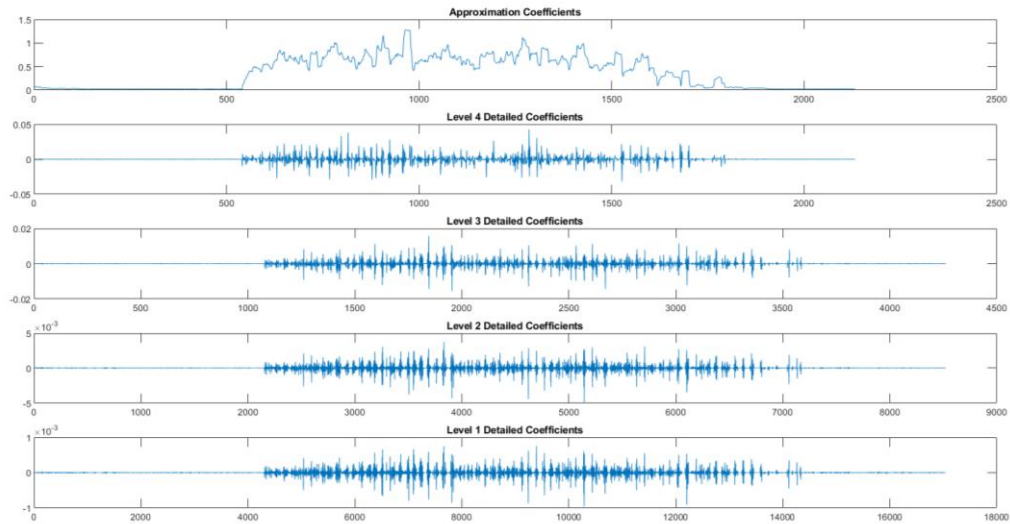


Figure 4.10: Multilevel decomposition for AD signal

4.4.2 Entropy

Entropy is the measure of information contained within a signal. It can be defined as an additive function where $E(0) = 0$ and

$$E(s) = \sum_i E(s_i) \quad \text{Eq. 4.2}$$

Where s_i represents each sample. The entropy values for the various decomposition values are as shown in Table 4.1.

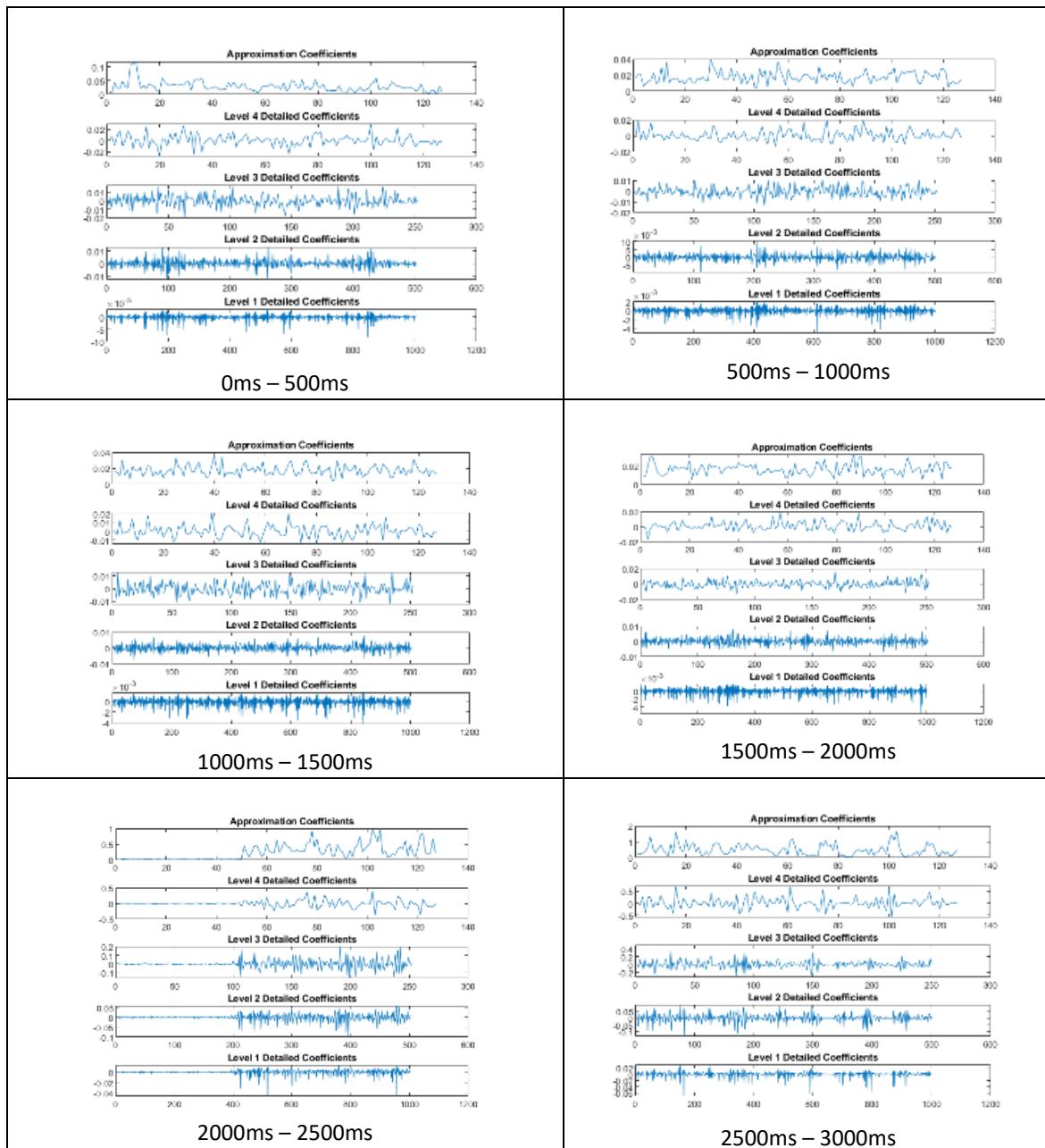
Table 4.1 Entropy value for entire duration

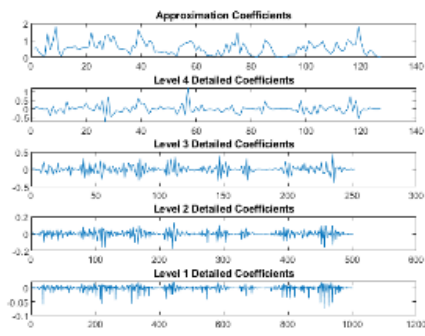
	AD	BR	FD	CED	FDI
CA1	343.7317	282.1905	142.6223	298.5561	144.2554
CD1	0.0017	0.00016805	0.00054175	0.00061189	0.0012
CD2	0.0182	0.002	0.0053	0.0064	0.0098
CD3	0.1298	0.0198	0.0290	0.0440	0.0430
CD4	0.4989	0.1347	0.0932	0.1433	0.1082

The values from the CA1 are able to indicate the activation of the various muscles in the arm. However, analysis of the entire duration is not suitable in a real-world application where the controlled device is required to react in real-time. Furthermore, a single-value attained across the entire duration of the trial is not able to distinguish between the various stages of the arm movement.

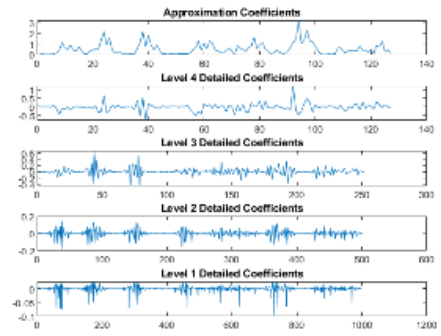
4.4.3 Real-Time Signal Processing

In order to be able to make decisions in real-time, the incoming data is segmented into windows of 500ms, giving us a total of 16 windows for the 8s data. Performing the DWT on the windows gives us the following waveforms for the AD Channel, as shown in Figure 4.11.

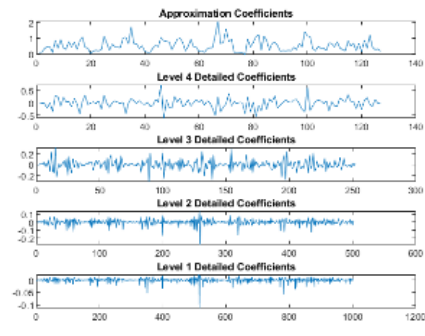




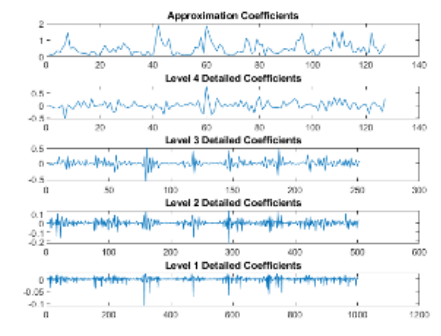
3000ms – 3500ms



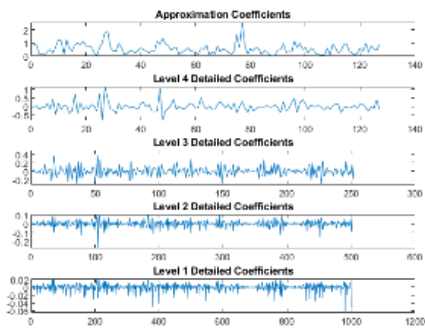
3500ms – 4000ms



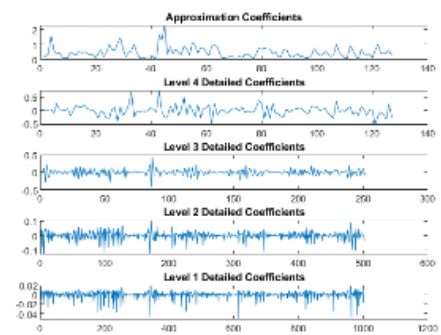
4000ms – 4500ms



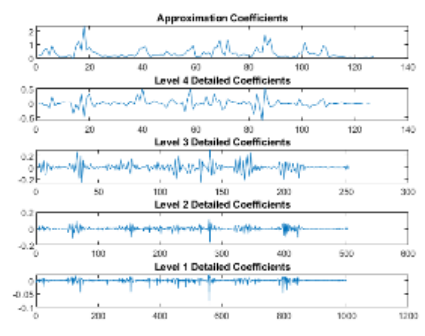
4500ms – 5000ms



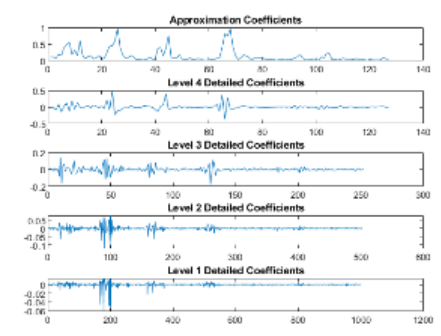
5000ms – 5500ms



5500ms – 6000ms



6000ms – 6500ms



6500ms – 7000ms

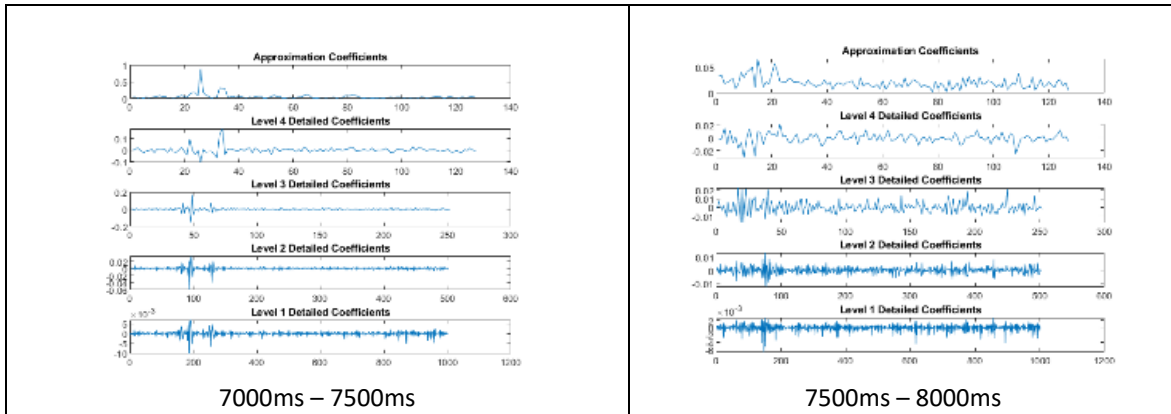


Fig 4.11: Wavelet Decomposition of AD over 16 windows (500ms each)

In the Tables 4.2 to 4.6, we examine the entropy values obtained for the various windows together with the corresponding arm movement.

Table 4.2: AD Windowed Entropy

State	(a)	(a)	(a)	(a)	(b)->(c)	(c)	(d)	(d)->(e)	Stay at (e)	Stay at (e)	(f) ->(g)	(g)	(g)	(g)->(h)	(h)	(h)
cA_4	0.91110	0.34336	0.33363	0.30034	16.36376	15.28840	7.29507	-47.8944	9.04632	4.08091	-2.30032	9.08831	0.34709	8.97632	2.71557	0.45164
cD_4	0.01679	0.00806	0.00823	0.01003	0.15978	0.50924	0.62912	0.75110	0.47001	0.65565	0.53610	0.41543	0.29886	0.13052	0.01531	0.01007
cD_3	0.06419	0.03648	0.03471	0.03476	0.99250	2.48889	3.50421	3.84979	3.25261	3.43541	3.59718	2.50890	1.72576	0.66125	0.12275	0.04553
cD_2	0.10285	0.05505	0.05598	0.05096	3.38998	8.37189	11.79911	9.76368	8.40861	11.04215	11.71331	8.11229	5.61222	1.46100	0.46619	0.08729
cD_1	0.10306	0.05079	0.05583	0.04421	4.98423	13.32184	11.55383	10.64924	11.41212	9.37020	12.99664	11.06937	6.77439	3.00313	0.49767	0.06618
Time(ms)	0 - 500	500 - 1000	1000 - 1500	1500 - 2000	2000 - 2500	2500 - 3000	3000 - 3500	3500 - 4000	4000 - 4500	4500 - 5000	5000 - 5500	5500 - 6000	6000 - 6500	6500 - 7000	7000 - 7500	7500 - 8000
Label	Rest				Forward		Grip	Lift	Stay		Down	Reverse		Rest		

Table 4.3: BR Windowed Entropy

State	(a)	(a)	(a)	(a)	(b)->(c)	(c)	(d)	(d)->(e)	Stay at (e)	Stay at (e)	(f) ->(g)	(g)	(g)	(g)->(h)	(h)	(h)
cA_4	5.16210	0.48592	0.37935	0.34845	7.01254	6.55060	14.50718	18.85079	18.02945	20.34798	20.01326	16.84822	14.05994	15.29040	12.40804	0.44155
cD_4	0.02013	0.01236	0.00850	0.00800	0.02486	0.02121	0.04023	0.04132	0.04779	0.03582	0.04096	0.04519	0.02842	0.04316	0.05630	0.01144
cD_3	0.10255	0.05297	0.03702	0.03985	0.16798	0.12961	0.29939	0.27845	0.33944	0.22466	0.22211	0.30674	0.18462	0.25694	0.27467	0.05237
cD_2	0.31602	0.08126	0.07307	0.05640	0.66897	0.44841	1.50759	1.32311	1.44717	1.19066	1.22457	1.27348	0.87457	1.12268	1.22657	0.07640
cD_1	0.67267	0.08583	0.06366	0.05911	1.88969	0.82140	5.03684	3.64853	4.49656	3.19691	3.49172	3.27551	3.00454	2.55175	2.48572	0.08067
Time(ms)	0 - 500	500 - 1000	1000 - 1500	1500 - 2000	2000 - 2500	2500 - 3000	3000 - 3500	3500 - 4000	4000 - 4500	4500 - 5000	5000 - 5500	5500 - 6000	6000 - 6500	6500 - 7000	7000 - 7500	7500 - 8000
Label	Rest				Forward		Grip	Lift	Stay		Down	Reverse		Rest		

Table 4.4: FD Windowed Entropy

State	(a)	(a)	(a)	(a)	(b)->(c)	(c)	(d)	(d)->(e)	Stay at (e)	Stay at (e)	(f) ->(g)	(g)	(g)	(g)->(h)	(h)	(h)
cA_4	1.37528	1.18288	0.98642	0.94881	2.70466	5.18911	11.66454	12.21466	9.22322	6.76625	4.68005	6.02627	10.86100	9.69764	1.34182	1.13031
cD_4	0.01749	0.01693	0.01655	0.01348	0.02612	0.05976	0.17893	0.17021	0.11332	0.06352	0.04525	0.06825	0.20985	0.24491	0.01224	0.01319
cD_3	0.07363	0.07708	0.06285	0.07167	0.14138	0.30867	1.10515	0.92275	0.63529	0.37314	0.26071	0.38374	0.98991	1.29853	0.06275	0.06143
cD_2	0.13235	0.15533	0.15394	0.18134	0.37379	0.76801	2.30035	2.83001	1.67197	1.50472	0.69083	1.18516	3.13005	3.23651	0.18147	0.18922
cD_1	0.19989	0.28159	0.18917	0.16091	0.40155	1.05962	3.04648	3.19856	2.45254	1.70379	0.71219	1.43605	3.04909	2.24181	0.21695	0.17910
Time(ms)	0 - 500	500 - 1000	1000 - 1500	1500 - 2000	2000 - 2500	2500 - 3000	3000 - 3500	3500 - 4000	4000 - 4500	4500 - 5000	5000 - 5500	5500 - 6000	6000 - 6500	6500 - 7000	7000 - 7500	7500 - 8000
Label	Rest				Forward		Grip	Lift	Stay		Down	Reverse		Rest		

Table 4.5: CED Windowed Entropy

State	(a)	(a)	(a)	(a)	(b)->(c)	(c)	(d)	(d)->(e)	Stay at (e)	Stay at (e)	(f)->(g)	(g)	(g)	(g)->(h)	(h)	(h)
cA_4	8.35882	5.33607	3.67097	3.92296	3.98115	12.82452	16.59909	18.97961	18.50493	18.27692	17.71798	18.90394	17.76524	13.32908	9.30891	2.44950
cD_4	0.08215	0.03392	0.03008	0.03250	0.04600	0.10603	0.22788	0.25735	0.23074	0.22295	0.19845	0.24297	0.27050	0.12679	0.08446	0.01610
cD_3	0.42425	0.21818	0.15764	0.18438	0.22622	0.64857	1.34424	1.21965	1.04731	1.21371	0.99124	1.43898	1.56317	0.71224	0.48718	0.09148
cD_2	1.09159	0.83345	0.72043	0.57446	0.53421	1.64606	3.92332	3.23371	3.34492	3.19086	3.13588	3.93788	4.99656	2.25924	1.55168	0.31778
cD_1	2.20308	1.59933	0.89519	1.16740	0.67816	2.76358	5.36669	4.27777	4.00822	4.17105	4.08770	5.15076	6.56204	3.03656	2.09344	0.56143
Time(ms)	0 - 500	500 - 1000	1000 - 1500	1500 - 2000	2000 - 2500	2500 - 3000	3000 - 3500	3500 - 4000	4000 - 4500	4500 - 5000	5000 - 5500	5500 - 6000	6000 - 6500	6500 - 7000	7000 - 7500	7500 - 8000
Label	Rest				Forward		Grip	Lift	Stay		Down	Reverse		Rest		

Table 4.6: FDI Windowed Entropy

State	(a)	(a)	(a)	(a)	(b)->(c)	(c)	(d)	(d)->(e)	Stay at (e)	Stay at (e)	(f)->(g)	(g)	(g)	(g)->(h)	(h)	(h)
cA_4	2.73608	0.82797	0.94037	0.61297	2.54765	5.10758	-0.65546	14.45859	12.08920	11.13030	11.19211	6.64524	4.75710	3.63076	1.34840	0.96100
cD_4	0.01569	0.01481	0.01511	0.01197	0.03125	0.15125	2.10429	0.51755	0.46759	0.27082	0.31516	0.18141	0.12184	0.09035	0.01699	0.01653
cD_3	0.08920	0.06495	0.07540	0.04870	0.17433	0.63757	7.94533	2.21787	1.96381	1.45338	1.37258	0.86377	0.52406	0.35296	0.08361	0.08073
cD_2	0.23332	0.14608	0.18607	0.10230	0.33671	1.03470	13.00275	4.08079	3.73848	2.80962	2.82053	1.34603	1.00400	0.51874	0.17312	0.18509
cD_1	0.28012	0.15379	0.15541	0.11508	0.42542	1.10977	7.78838	3.22486	3.25561	2.05462	1.95008	1.08259	0.90156	0.43353	0.15355	0.12566
Time(ms)	0 - 500	500 - 1000	1000 - 1500	1500 - 2000	2000 - 2500	2500 - 3000	3000 - 3500	3500 - 4000	4000 - 4500	4500 - 5000	5000 - 5500	5500 - 6000	6000 - 6500	6500 - 7000	7000 - 7500	7500 - 8000
Label	Rest				Forward		Grip	Lift	Stay		Down	Reverse		Rest		

The states represent the position and movement of the arm during the various stages of the trial. They can be translated to the labels shown in the last row of each table. These labels will be used in the training and testing of the ML model.

4. 5 Machine Learning & Classification

4.5.1 Data Preparation

The data is first prepared by arranging the data and labels according to the specific classification we wish to perform. At the first level, a classification of both the ‘Rest’ and ‘Active’ states is performed as this is critical in the application of such systems in a natural setting. We will take 1s data of the ‘Rest’ state and 1s data from the ‘Active’ state for each of the EMG sensors. The label set will also be prepared to represent the states. Table 4.7 shows the data for the AD channel.

Table 4.7: Input Data and Label for ‘Rest’ and ‘Active’ State

Input	cA_4	0.33363	0.30034	16.36376	15.28840
	cD_4	0.00823	0.01003	0.15978	0.50924
	cD_3	0.03471	0.03476	0.99250	2.48889
	cD_2	0.05598	0.05096	3.38998	8.37189
	cD_1	0.05583	0.04421	4.98423	13.32184
Output Label		1	1	0	0
		0	0	1	1

4.5.2 Neural Network

A feedforward neural network of 3 layers and 1 hidden layer is first created. The hidden layer is programmed to have 10 neurons. The layout is as shown in Fig. 4.12.

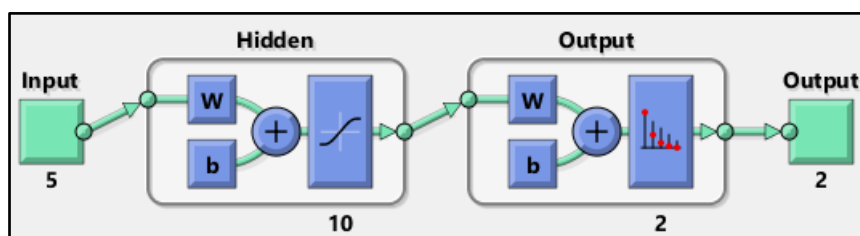
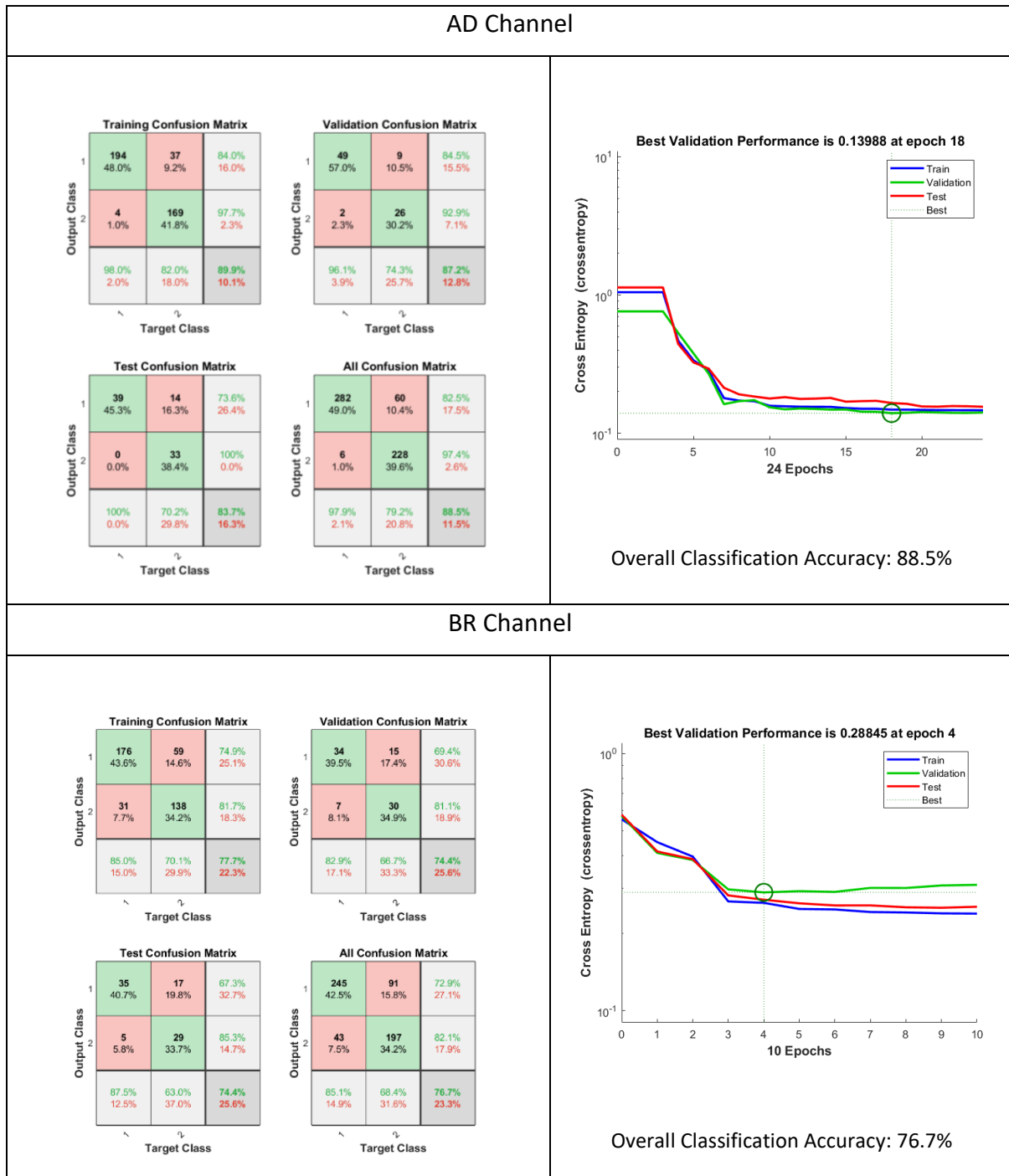


Fig. 4.12: ANN Model

The entire dataset is split into 3 sections: 70% for Training, 15% for Validation and 15% for Testing. The results for the various channels are shown in Table 4.8.

Table 4.8 Classification Accuracy for Individual Channels



FD Channel

Training Confusion Matrix

Output Class	1	161 39.9%	92 22.8%	63.6% 36.4%
	2	48 11.9%	103 25.5%	68.2% 31.8%
		77.0% 23.0%	52.8% 47.2%	65.3% 34.7%
		Target Class		

Validation Confusion Matrix

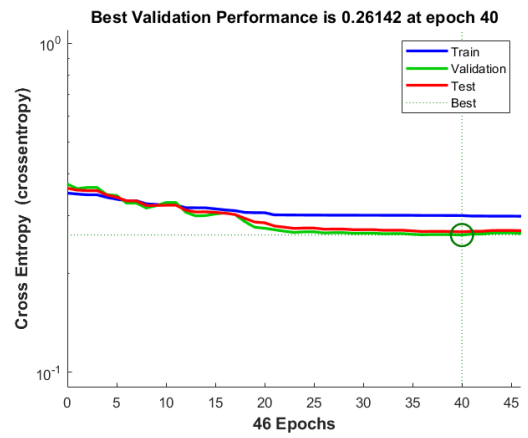
Output Class	1	29 33.7%	19 22.1%	60.4% 39.6%
	2	7 8.1%	31 36.0%	81.6% 18.4%
		80.6% 19.4%	62.0% 38.0%	69.8% 30.2%
		Target Class		

Test Confusion Matrix

Output Class	1	38 44.2%	21 24.4%	64.4% 35.6%
	2	5 5.8%	22 25.6%	81.5% 18.5%
		88.4% 11.6%	51.2% 48.8%	69.8% 30.2%
		Target Class		

All Confusion Matrix

Output Class	1	228 39.6%	132 22.9%	63.3% 36.7%
	2	60 10.4%	156 27.1%	72.2% 27.8%
		79.2% 20.8%	54.2% 45.8%	66.7% 33.3%
		Target Class		



Overall Classification Accuracy: 66.7%

CED

Training Confusion Matrix

Output Class	1	185 45.8%	56 13.9%	76.8% 23.2%
	2	21 5.2%	142 35.1%	87.1% 12.9%
		89.8% 10.2%	71.7% 28.3%	80.9% 19.1%
		Target Class		

Validation Confusion Matrix

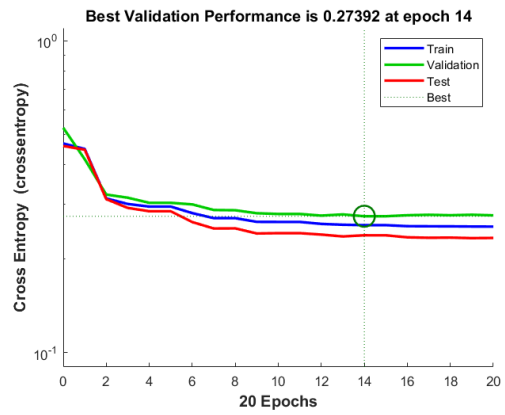
Output Class	1	32 37.2%	19 22.1%	62.7% 37.3%
	2	7 8.1%	28 32.6%	80.0% 20.0%
		82.1% 17.9%	59.6% 40.4%	69.8% 30.2%
		Target Class		

Test Confusion Matrix

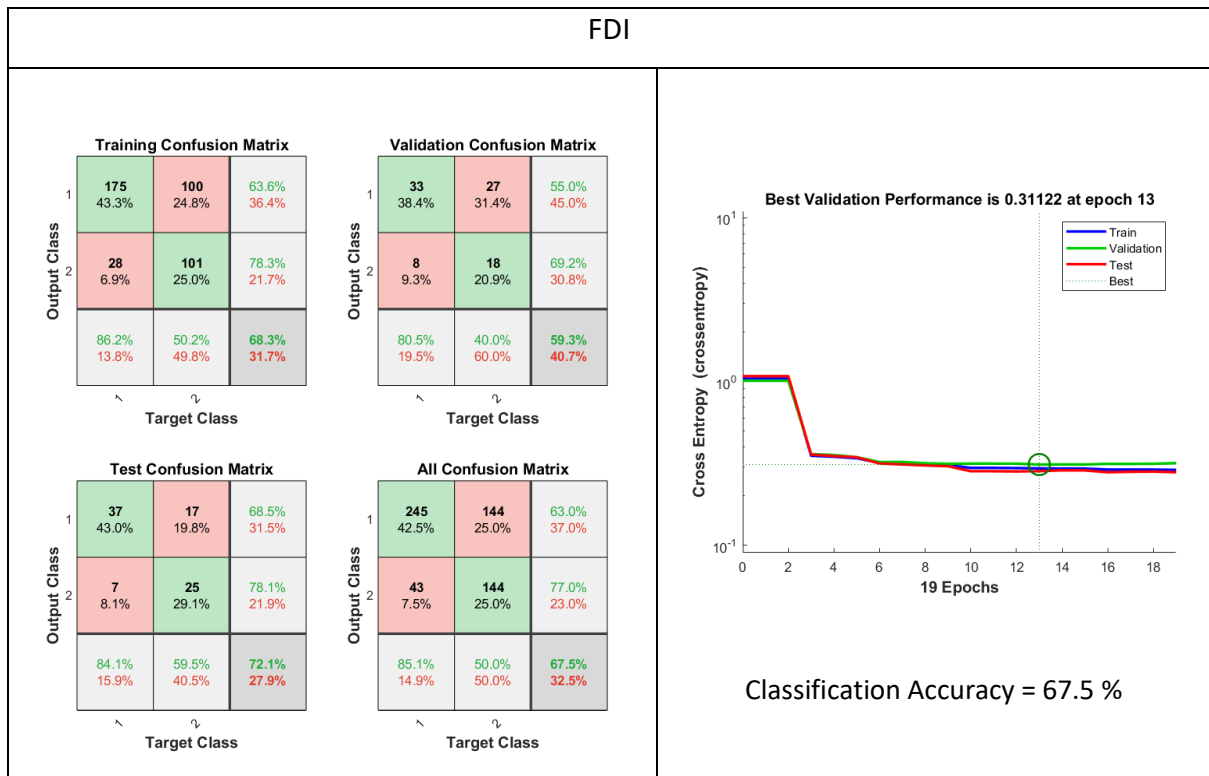
Output Class	1	40 46.5%	10 11.6%	80.0% 20.0%
	2	3 3.5%	33 38.4%	91.7% 8.3%
		93.0% 7.0%	76.7% 23.3%	84.9% 15.1%
		Target Class		

All Confusion Matrix

Output Class	1	257 44.6%	85 14.8%	75.1% 24.9%
	2	31 5.4%	203 35.2%	86.8% 13.2%
		89.2% 10.8%	70.5% 29.5%	79.9% 20.1%
		Target Class		



Overall Classification Accuracy: 79.9%



It can be seen that the AD Channel has the highest classification accuracy of 88.5%. As the AD channel corresponds to the shoulder area, it naturally has a great deal of activity when the arm starts to move forward from the rest state. A similar approach is taken to differentiate the Forward / Reverse actions and the Raise/Lower actions of the arm. The results are as shown below in Table 4.9.

Table 4.9: Classification Accuracy Summary

EMG Channel	Forward / Reverse	Raise / Lower
AD	78.9%	87.9%
BR	76.5%	61.3%
FD	73.1%	62.1%
CED	52.9%	77.8%
FDI	49.5%	83.5%

The raise and lower actions can be accurately classified using the AD, CED and FDI channels, while the forward and reverse actions can be classified using the AD, BR and FD channels.

4.6 Unified Model

In order to improve the accuracy of EMG-based classification, the correct sensor signals should be used for the various actions. The proposed unified EMG model, as shown in Fig. 4.13, utilizes this approach to deliver high classification accuracies.

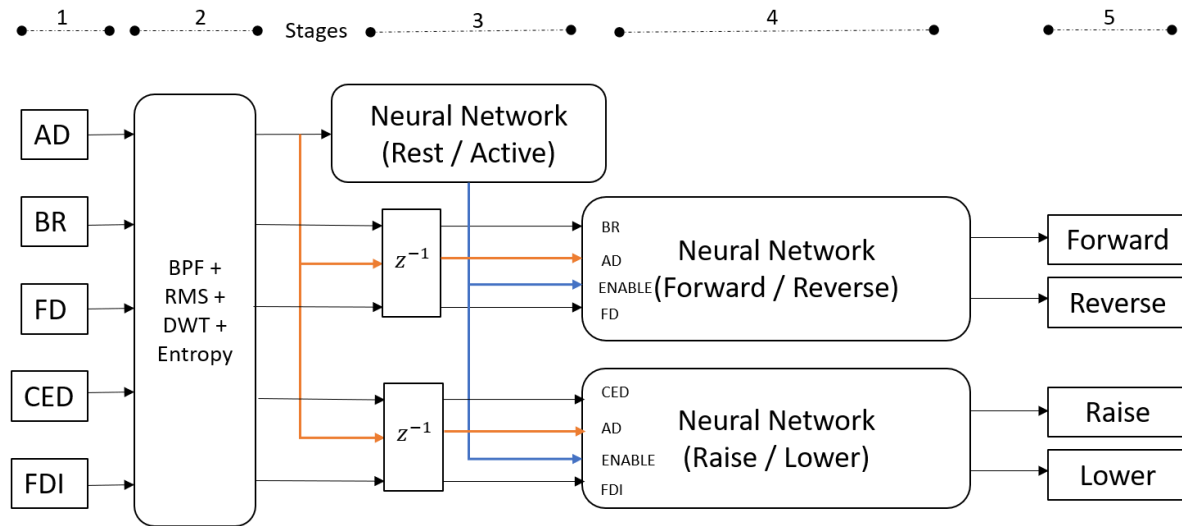


Fig 4.13: Unified EMG Classification Model

The model has five stages.

- Stage 1

In this stage the raw EMG data is captured using appropriate sensor hardware.

- Stage 2

Here, the Band-Pass filtering, RMS feature extraction, DWT coefficient computation and Entropy calculations are performed.

- Stage 3

In Stage 3, the Network to classify Rest/Active is first utilised. The output of this network is then used to 'Enable' the two other Networks only if it classified the data as representing the 'Active' state. This would help prevent unnecessary computation during the 'Rest' state of the system.

- Stage 4

Here, two separate networks process the features in parallel. The network for the Forward and Reverse classification received the BR, AD and FD features while the network for the raise and lower classification receives the CED, AD and FDI. Both these networks receive these

features with a 500ms delay (window size). This allows the outcome from Stage 3 and its associated data to arrive at the same time in Stage 4.

- Stage 5

In this final stage, the outputs are updated according to the received data.

4.7 Results & Discussion

From Table 4.9, we can see that the accuracy for the Rest/Active classification remains the same as before. This is because the same AD channel is used for the network. The forward/reverse accuracy has increased from 78.9% to 91.9%, while the raise/lower accuracy has increased from 87.9% to 93.5%.

Table 4.9: Classification Accuracy of Unified Model

State	Accuracy
Rest / Active	88.5%
Forward / Reverse	91.9%
Raise / Lower	93.5%

The increase in the classification accuracy is due to networks being trained with a synergy of channels that presented the highest accuracy in the individual stages. The proposed model is able to decode the selected channels based upon the active state of the sensors.

4.8 Conclusion

A comparison of state-of-the-art is shown in Table 4.10. In [208], the authors used Deep Learning to classify open/close and various finger movements. An accuracy of 93% was achieved, though their approach was computationally complex and required the use of 2 EMG channels. In [209], the use of Temporal Convolutional Networks (TCN) for up to 8 EMG channels was investigated for hand and finger movements, yielding accuracy results of 89.76%. [210] investigated the use of a hybrid Convolutional Neural Network (CNN and Recurrent Neural Network (RNN) on different datasets and achieved accuracies ranging from 87% to 99%.

Table 4.10: Comparison of State-of-the-Art

Ref.	Number of Channels	Accuracy
[208]	2	93%
[209]	8	89.76%
[210]	8 – 16	87% - 99%

The proposed model is able to deliver a high level of accuracy using only 5 EMG channels. The layered approach allows the use of specific EMG channels for targeted activities. It also provides a means of putting the unused blocks in a low-power mode when the system is in the 'Rest' state. The results are promising and the future work of this would be the implementation of the proposed algorithm on a stand-alone embedded board.

Now that we have obtained a good understanding of these physiological signals and their associated signal processing and classification techniques, we can go deeper into developing a more robust architecture that more efficiently understands these signals.

Publication:

Motor State Classification based on Electromyography (EMG) Signals using Wavelet Entropy and Neural Networks

2021 International Conference on Computational Science and Computational Intelligence (CSCI) 2021-12 | Conference paper

DOI: [10.1109/csci54926.2021.00248](https://doi.org/10.1109/csci54926.2021.00248)

CONTRIBUTORS: Ravi Suppiah; Khalid Abidi; Noori Kim; Anurag Sharma

Chapter 5

FIS-LSTM Network

In Chapters 3 and 4, we performed signal processing and machine learning on EEG and EMG signals. The ML techniques at that time used data that was sampled at specific instances of time and analyzed how the extracted features represented the intention of the user. To be able to make sense of these feature changes over a time-interval, a more holistic approach would be to analyze the data over the timeframe of the entire action. In this chapter we will apply this approach to both EEG and EMG signals.

5.1 EEG ANALYSIS

EEG signals exhibit both time and frequency changes in specific electrode regions associated with the type of activity or stimulus. To fully understand the changes in these signals, we need to be able to analyze its change over a subset of time. This requires a solution that has the ability to remember and detect specific patterns in the signals over time. The Long Short-Term Memory (LSTM) is a network that is able to achieve this objective. It has been utilized in the analysis of various time-series data which are representative of many biological signals [211, 212]. LSTM has also been utilized in many other applications that require memory and the “context” of the data with respect to its past. Natural Language Processing (NLP) is such an example.

LSTM Architecture

An LSTM network, as shown in Figure 5.1, is a form of Recurrent Neural Network (RNN). RNN's are able to effectively learn short-term dependencies but they have an issue with long-term

dependencies. LSTM's were designed to address the issue so that it can effectively learn both short and long term patterns in the data [213].

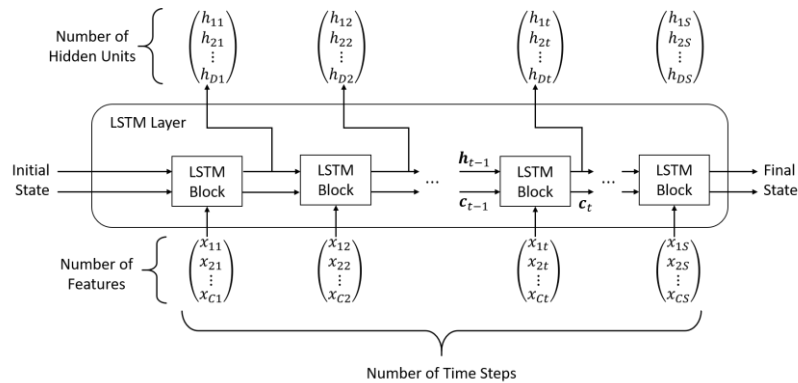


Fig 5.1: LSTM Layer Architecture

Figure 5.2 illustrates the manner in which data flows through a LSTM layer. It shows a time series X , with C features (channels) with a length of S . The cell state at any time step t , is denoted by both h_t and c_t . The various cell-state components are described in Table 5.1.

The initial state of the LSTM block is used for the first LSTM block, together with the first-time step of the feature sequence. At any time-step t , there is a new set of inputs. Each LSTM block uses the current network state (c_{t-1}, h_{t-1}) to compute the output and provided the updated cell state c_t . The hidden state (output state) at time step t , provides the LSTM layer output for that particular time step. Information learned from previous time steps is contained within the cell state. The cell state is constantly updated to either add or remove information from it. This is achieved through the use of Gate. The hidden state and the cell state of each layer are controlled by the following components.

Table 5.1: Cell State Components of LSTM Network

Component	Purpose
Input Gate (i)	Control level of cell state update
Forget Gate (f)	Control level of cell state reset (forget)
Cell Candidate (g)	Add information to cell state
Output Gate (o)	Control level of cell state added to hidden state

The flow of data at a particular time step t , is shown below.

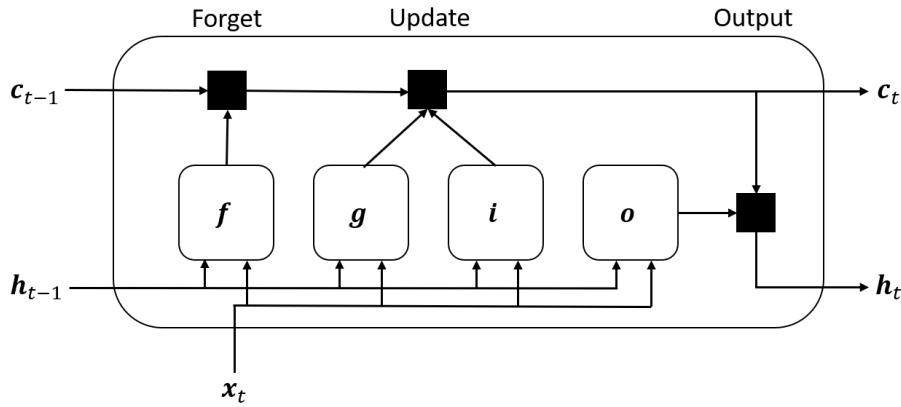


Fig 5.2: LSTM Cell

The input weights W , the recurrent weights R , and the bias b , are the learnable weights of the LSTM layer. These inputs are concatenated to form the following matrices W, R and b . Further concatenation of the matrices give us the following:

$$W = \begin{bmatrix} W_i \\ W_f \\ W_g \\ W_o \end{bmatrix}, R = \begin{bmatrix} R_i \\ R_f \\ R_g \\ R_o \end{bmatrix}, b = \begin{bmatrix} b_i \\ b_f \\ b_g \\ b_o \end{bmatrix}, \quad \text{Eq. 5.1}$$

At any time, the state of the cell is given by

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \quad \text{Eq. 5.2}$$

The hidden state is computed as

$$h_t = o_t \odot \sigma_c(c_t) \quad \text{Eq. 5.3}$$

The state activation function is represented by σ_c . The LSTM function uses the hyperbolic tangent function (tanh) to compute the state activation function.

The formulae in Table 5.2 are used to compute the components at time t .

Table 5.2: Activation Formulae for Cell State Components of LSTM Network

Component	Formula	
Input Gate (i)	$i_t = \sigma_g(W_i x_t + R_i h_{t-1} + b_i)$	Eq. 5.4
Forget Gate (f)	$f_t = \sigma_g(W_f x_t + R_f h_{t-1} + b_f)$	Eq. 5.5

Cell Candidate (g)	$g_t = \sigma_c(W_g x_t + R_g h_{t-1} + b_g)$	Eq. 5.6
Output Gate (o)	$o_t = \sigma_g(W_g x_t + R_g h_{t-1} + b_g)$	Eq. 5.7

The gate activation function σ_g uses the sigmoid function defined as

$$\sigma(x) = (1 + e^{-x})^{-1} \quad \text{Eq. 5.8}$$

Dataset

The authors have used the dataset provided in [188] in this research, as described in Section 3.2

Data Preparation

The chosen dataset has a wide variety of 'Series' that allows the participants to use different weights with the same surface (weight series) or use the same surface with different weights (surface series). There are 3 different sets of weight used, 165g, 330g and 660g. In the following analysis, three series from each participant using the 330g weight is extracted. A single weight is chosen to ensure that the experimental setup is consistent across all the participants. In each series, there are 10 lifts with the expected weight. Each lift starts with a Rest period of 2s, followed by a forward movement of 2s. The action is triggered when the participant observes an LED being ON. The weight is held in a fixed position until the LED goes OFF. After that, the weight is lowered, and the arm returns to its original position. All the participants performed their trials using the Right Arm. It corresponds with the C3 channel of the EEG electrode placement. The remaining analysis will be performed using this channel alone to reduce the system's overall complexity. This analysis focuses on using the EEG signals to decode the arm's FOUR different actions: Forward, Grip & Lift, Down & Release, Reverse. The time-lines of these events can be seen in Figure 5.3.

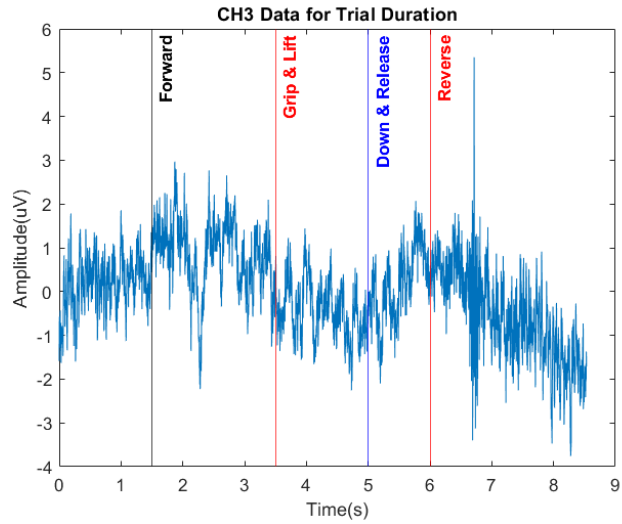


Figure 5.3: CH3 Data for Trial Duration with Event Markers

For the Forward and Reverse actions, 2s of data was used in the analysis, while for the Grip and Release actions, 0.5s of data was selected. Figures 5.4-5.5 show the overlapped signals for these two event pairs.

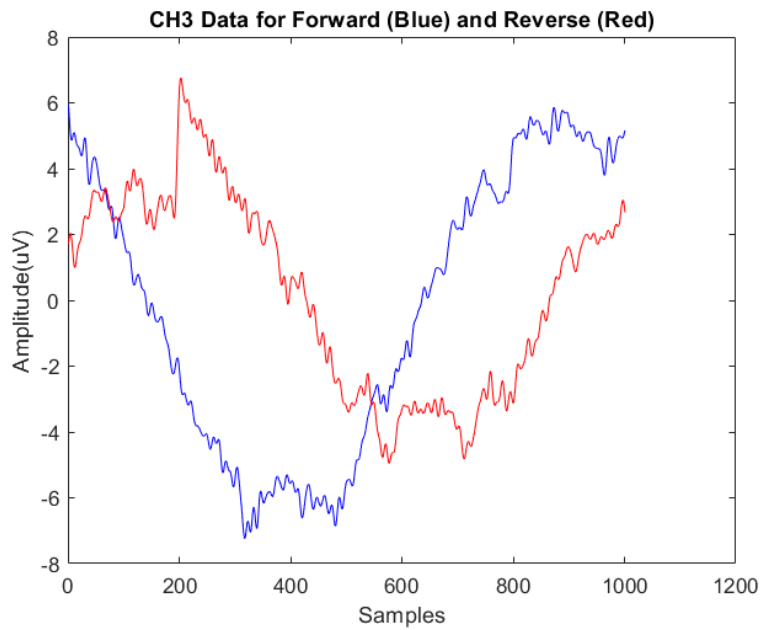


Fig 5.4: CH3 Data for the Forward (Blue) and Reverse (Red) movements

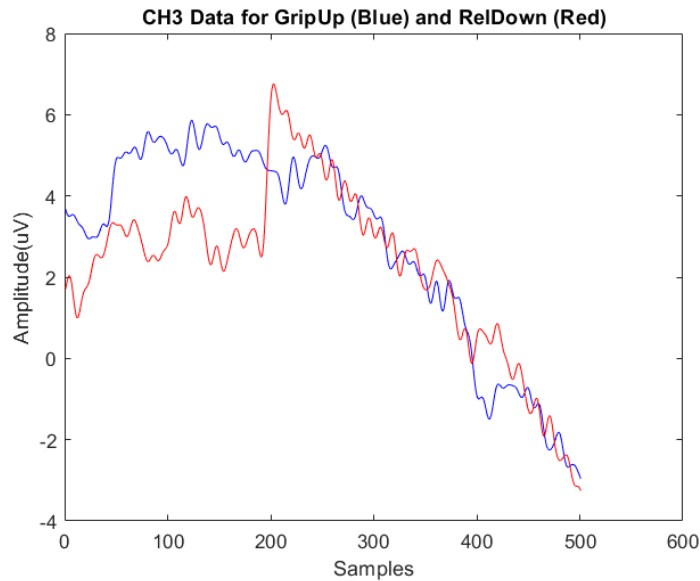


Fig 5.5: CH3 Data for the Grip_Up (Blue) and Release_Down (Red) movements

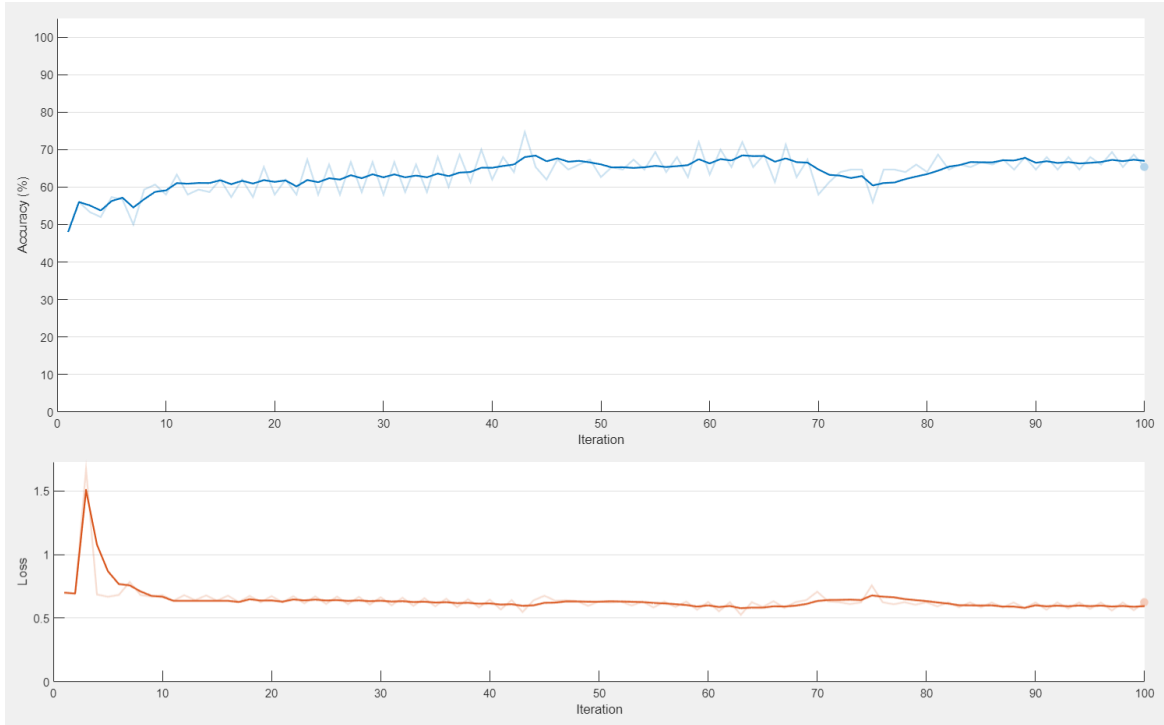
The above data pairs are extracted for all the trials for all 12 participants and arranged into cells as required by the *lstm* MATLAB library. In the first phase, the normalized time domain data is used to train the LSTM network.

5.2 LSTM Training and Testing

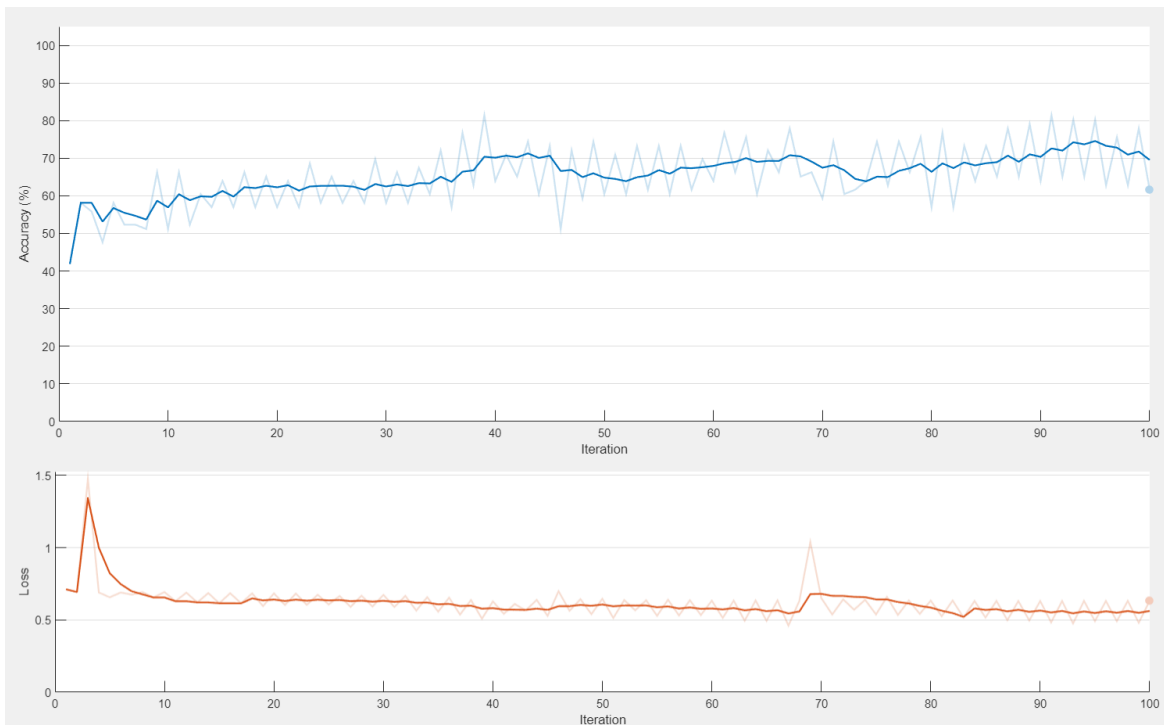
The LSTM Network was configured to have 100 cells with two fully connected layers. The processed data was split into 70% Training and 30% Testing sets. Both training and testing was done for 50 epochs with 10-fold cross validation being performed. Accuracy, Recall, Precision and F1-Score were computed based on the True Positive (TP), True negative (TN), False Positive (FP) and False Negative (FN) values.

Forward – Reverse Analysis

Figure 5.6 shows the Training and Testing Phase for the LSTM Network.



(a) Training Phase



(b) Testing Phase

Fig 5.6: LSTM performance for Training (a) and Testing (b) Phase

The confusion matrix for the training and testing phases are shown in Figure 5.7.

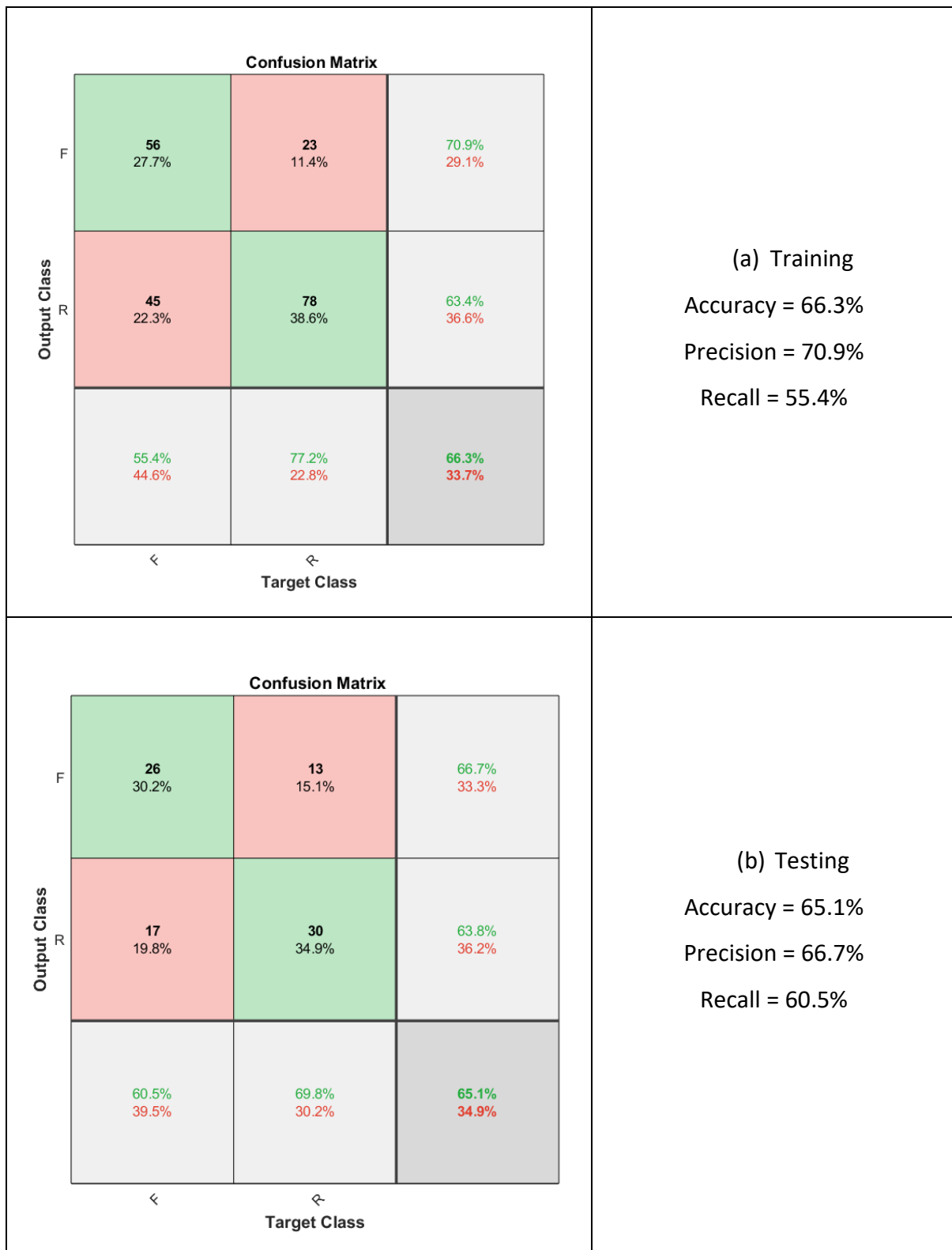


Fig 5.7: Confusion Matrix for Training (a) and Testing (b)

As can be seen, the classification accuracy based on raw time domain data is quite low, in the range of 65.1% - 66.3%.

Grip Up – Release Down Analysis

Figure 5.8 shows the Training and Testing Phase for the LSTM Network.



Fig 5.8: LSTM performance for Training (a) and Testing (b) Phase

The confusion matrix for the training and testing phases are shown in Figure 5.9.

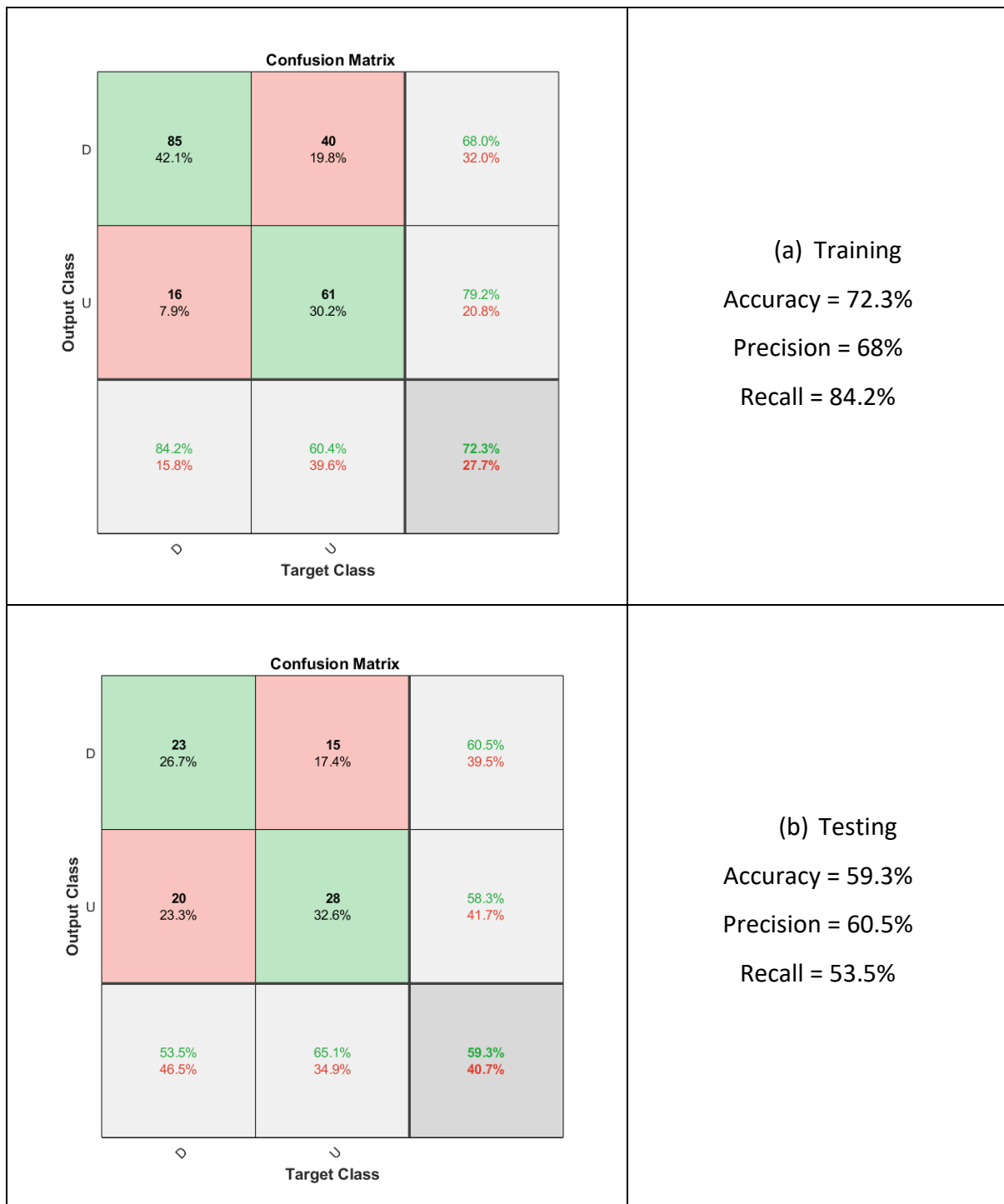


Fig 5.9: Confusion Matrix for Training (a) and Testing (b)

Time and Frequency Features

Table 5.3 shows the formulae for the time and frequency domain features that were computed.

Table 5.3: Time and Frequency Domain Features

Feature	Formula		Symbols
Mean	$\mu = \frac{1}{N} \sum_{i=1}^N x_i$	Eq. 5.9	N: Number of Samples
Skewness	$S = \frac{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^3}{\left(\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu)^2\right)^{3/2}}$	Eq. 5.10	x_i : Each Sample μ : Mean f : frequency
Power Spectrum Density (PSD)	$P(f) = \frac{\Delta t}{N} \left \sum_{n=0}^{N-1} x_n e^{-j2\pi f \Delta t n} \right ^2$	Eq. 5.11	Δt : Sample Interval

The PSD was computed for the various bands, Delta (0.5 – 4Hz), Theta (4 – 8Hz), Alpha (8 – 12Hz), Beta (12 – 30Hz), and Gamma (30 – 50Hz). The Forward and Reverse movements had 2s of data each, while the Grip and Release movements each had 1s of data each. Table 5.4 shows the various testing results obtained for the features using various window sizes ranging from 50ms to 250ms after performing 10-fold cross-validation with the dataset. The highest classification accuracy of 73.8±1.2%, as highlighted in the table, was obtained for the Alpha PSD feature with a 50ms window.

Table 5.4: LSTM Classification accuracy (Forward & Reverse) for different window size

Feature	10ms	25ms	50ms	100ms	150ms	200ms
Time-Series	60.2±1.2%	59.3±1.3%	61.5±3.1%	59.7±1.2%	54.1±1.2%	49.3±1.6%
Mean	61.7±1.1%	61.3±1.9%	60.3±2.1%	58.5±1.5%	52.3±1.9%	48.9±1.5%
Skewness	60.3±2.1%	60.4±1.8%	60.7±1.5%	55.3±1.9%	50.2±1.3%	48.3±1.4%
Theta PSD	51.7±1.2%	53.3±2.5%	58.2±1.3%	52.4±1.2%	51.9±2.3%	50.7±1.9%
Alpha PSD	62.2±1.7%	67.3±1.7%	73.8±1.2%	64.9±2.2%	57.3±1.7%	53.1±1.2%
Beta PSD	59.2±1.2%	59.9±2.3%	68.5±1.9%	54.6±2.2%	52.8±1.8%	49.5±1.6%
Gamma PSD	57.7±1.7%	56.2±1.6%	58.9±2.2%	51.8±1.9%	45.9±1.3%	45.5±1.7%

Figure 5.10 shows the PSD values for the different windows for the Alpha Channel. From the results in Table 5.4, it can be observed that the accuracy is highest for the Alpha PSD channel for a 50ms window. As the accuracy is highest for the 50ms window, PSD analysis will focus on this window size.

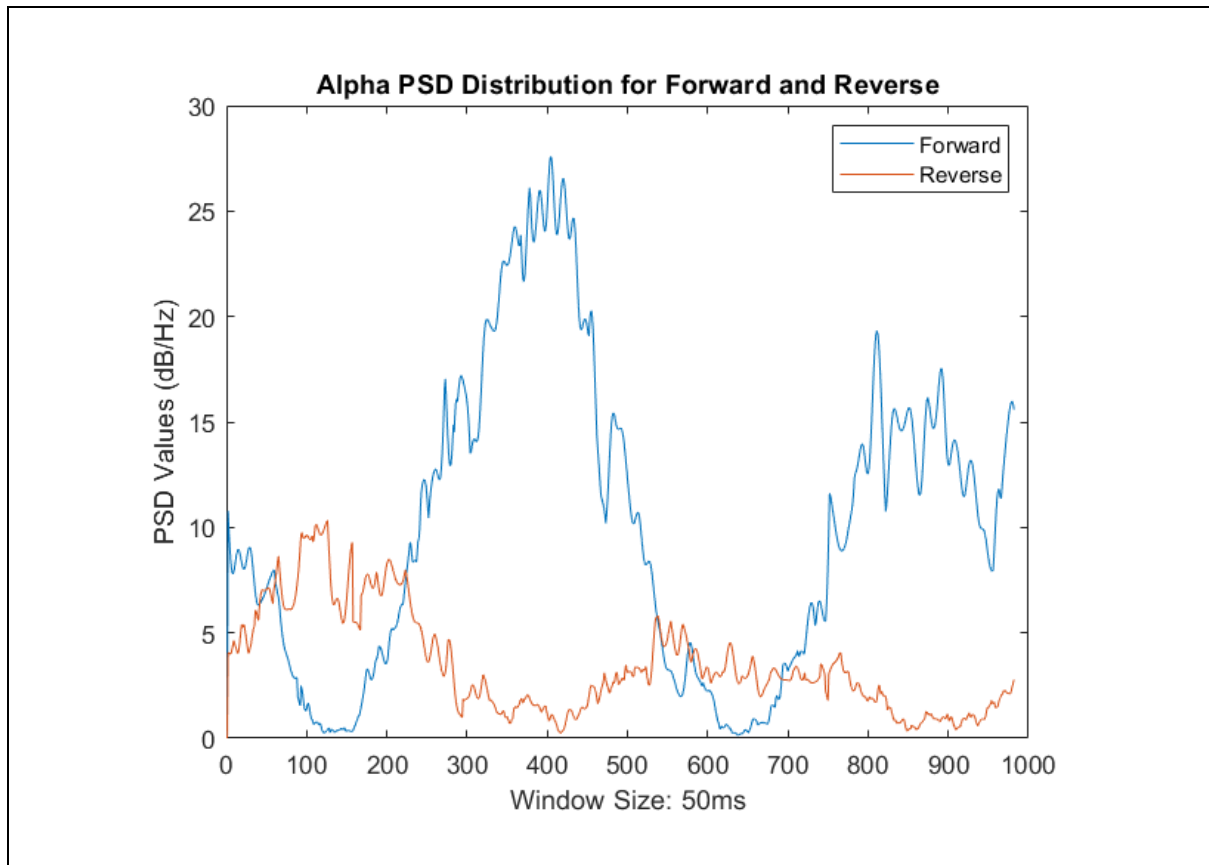


Figure 5.10: Alpha PSD Distribution for 50ms Window

Table 5.5: LSTM Classification accuracy (Grip_Up & Release_Down) for different window size

Feature	10ms	25ms	50ms	100ms	150ms	200ms
Time-Series	51.4±2.4%	50.1±1.3%	53.2±1.8%	47.2±1.9%	45.3±1.2%	39.5±2.2%
Mean	55.3±1.1%	56.1±1.9%	50.9±2.1%	43.5±1.5%	40.2±1.9%	38.1±1.5%
Skewness	59.5±2.1%	54.5±1.8%	51.2±1.5%	45.6±1.9%	41.2±1.3%	38.3±1.4%
Theta PSD	60.5±1.2%	61.3±2.5%	56.4±1.3%	51.7±1.2%	48.2±2.3%	42.7±1.9%
Alpha PSD	60.5±1.7%	61.8±1.7%	69.3±1.2%	50.2±2.2%	47.1±1.7%	43.1±1.2%
Beta PSD	53.2±1.2%	54.4±2.3%	61.2±1.9%	49.4±2.2%	45.2±1.8%	42.5±1.6%
Gamma PSD	50.1±1.7%	51.2±1.6%	56.8±2.2%	48.2±1.9%	43.2±1.3%	41.5±1.7%

The highest classification accuracy of $69.3 \pm 1.2\%$, as highlighted Table 5.5, was obtained for the Alpha PSD feature with a 50ms window. However, the classification accuracy is low and more analysis needs to be done. Figure 5.11 shows the Alpha distribution for the 50ms window.

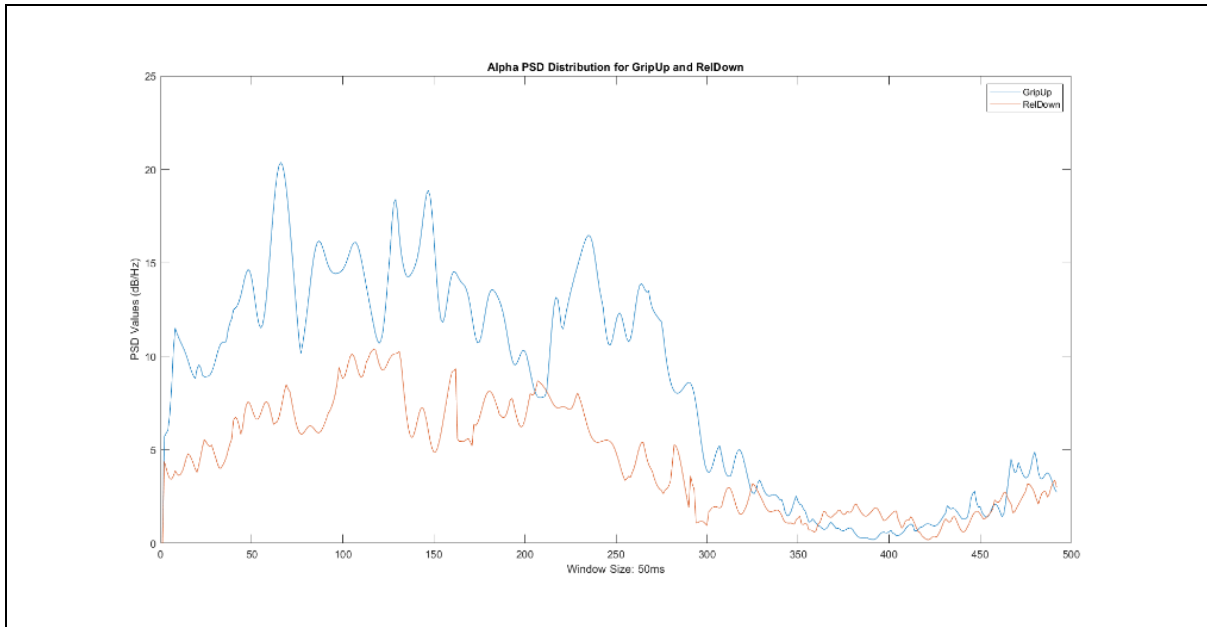


Figure 5.11: Alpha PSD Distribution for 50ms Window

It can be observed from Figure 5.11, that the GripUp and RelDown movements have a similar PSD distribution through time. This explains the low classification accuracy.

Further Analysis of PSD Bands

We can observe the Alpha and Beta Bands overlay in Figure 5.12.

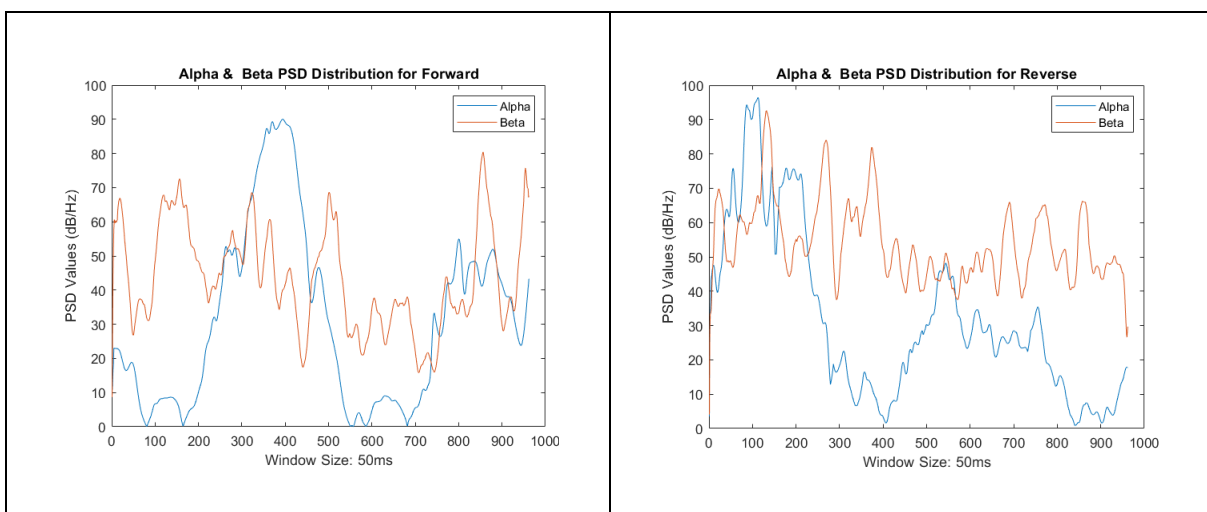


Figure 5.12: Alpha and Beta Overlays for Forward and Reverse Action

The key step would be to obtain a unified value that uses both the Alpha and Beta PSD values for each movement. This brings us to the next phase of this research.

5.3 Fuzzy Logic

Fuzzy Logic allows us to relate classes of objects without clearly defining any boundaries. It classifies objects based on membership functions that are formed as a matter of degree. It can also be viewed as a methodology for interpreting descriptive words rather than numbers, and in this manner, it is able to very closely approximate human reasoning. This give a very good balance between significance and trade-off.

Fuzzy Sets and Membership Functions

The basic idea that drives fuzzy systems are fuzzy sets. In normal set theory, elements that are assigned a value of 1 are interpreted to be a belonging to the set X , while those elements with a value of 0 do not belong to set X .

$$\mu_a = X \rightarrow \{0,1\} \quad \text{Eq. 5.12}$$

In a fuzzy set, we are able to add flexibility to the rules and this allows a more human-like interpretation of the data. This can be achieved by allowing values in-between 0 and 1.

$$\bar{A} = \{(x, \mu_a(x)) | x \in U\} \quad \text{Eq. 5.13}$$

Where $\mu_a(x)$ is a membership function (MF) of \bar{A} , where \bar{A} is a fuzzy set in the universe U . The membership function is what determines the degree of participation of each input. Each input has a weighted value which ay have some overlap with other inputs. The output value is finally determined using a set of rules that interpret these inputs. Some examples of Membership Functions (MF) are shown in the Figure 5.13.

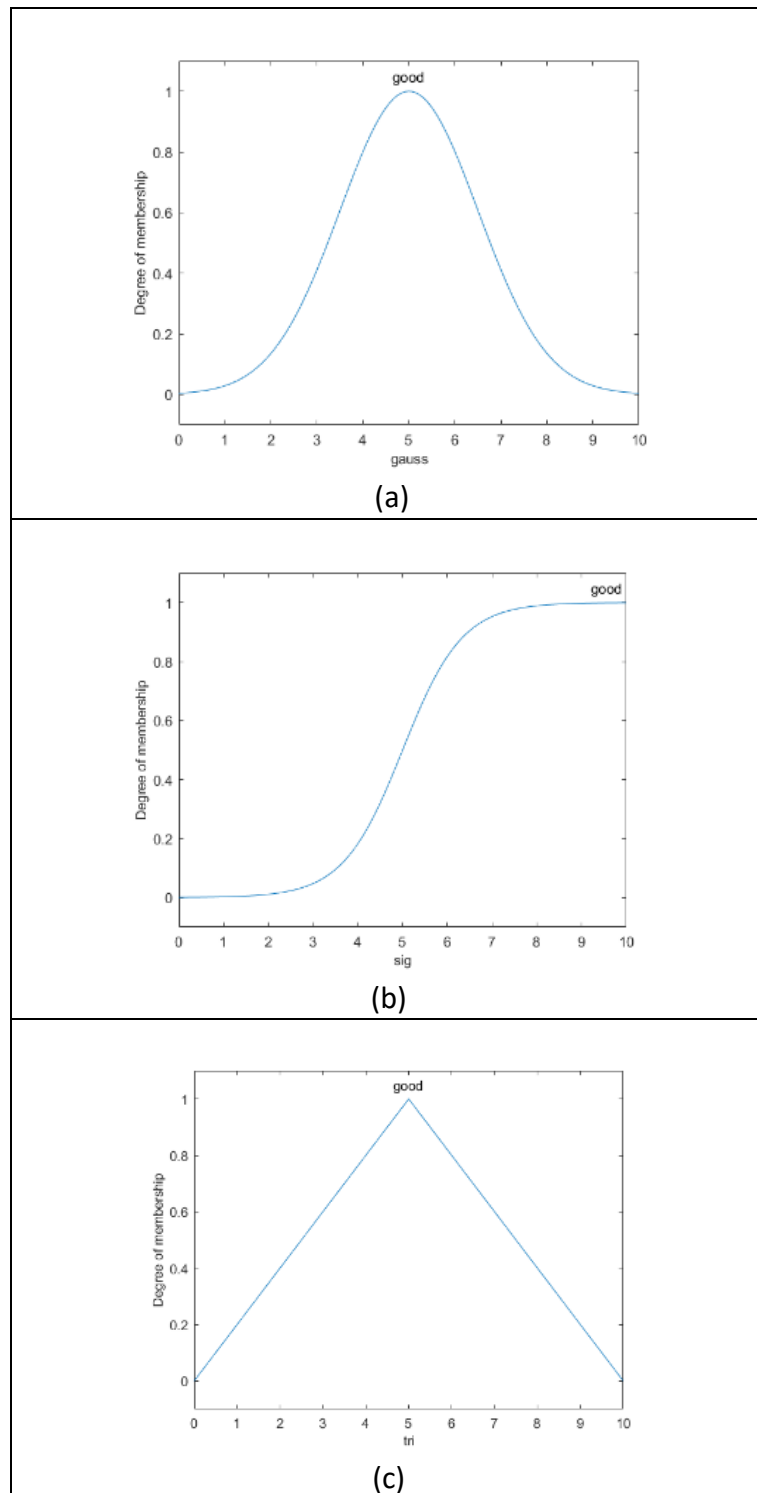


Figure 5.13: Types of Membership Function: (a) Gaussian, (b) Sigmoid, (c) Triangular

Fuzzy Operations

As shown in Figure 5.14, we can also perform Logical Operations such as, AND, OR and NOT, on Fuzzy Sets. The same logical concepts as digital logic applies, except that now we apply it to the MFs.

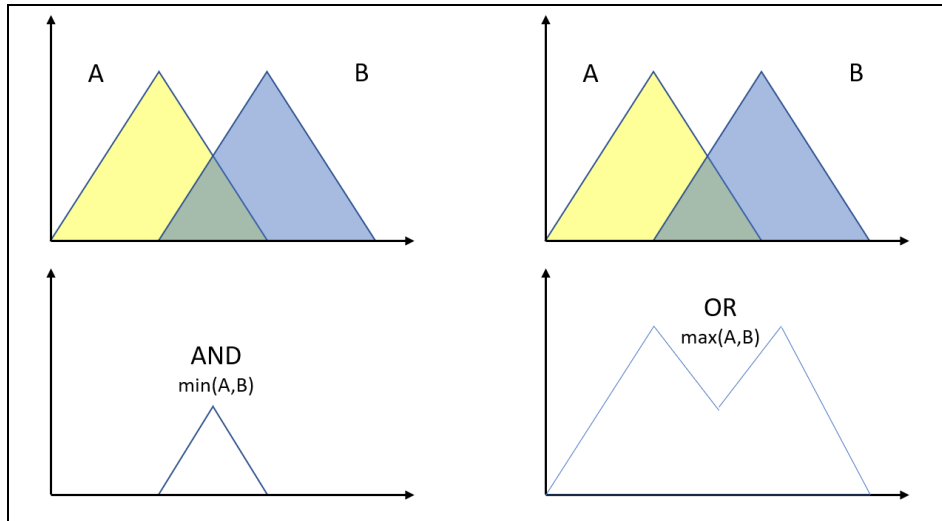


Figure 5.14: AND and OR operators for MFs

Fuzzy Classification

Fuzzy Classification is performed using a set of if-then rule statements that make use of fuzzy logic. In its simplest form, you can take it as such:

If x is A, then y is B

You can add many fuzzy inputs and multiple rules to the system.

If x is A and z is G then y is B

There are two-steps to interpret the if-then rules.

Step 1: Antecedent Evaluation: This involves fuzzifying the inputs and applying the necessary fuzzy operators

Step 2: Applying the result to the consequent

Defuzzification

The fuzzy inputs and the membership functions generate a range of output values that encompasses all of them. This must be defuzzified to obtain a single output value. There are many techniques available, and the most commonly used one is the centroid method. As

shown in Figure 5.15, This approach computes the center of the area of the whole area under the combined fuzzy set output.

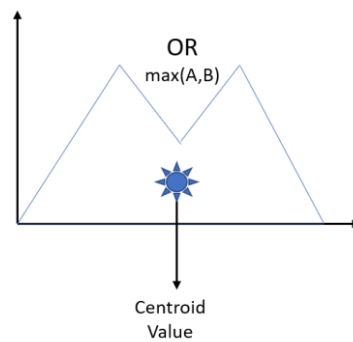


Figure 5.15: Defuzzification using Centroid Method

5.4 Applying Fuzzy Logic to EEG Signals

Observation of the Alpha and Beta PSD Signals for the Forward and Reverse Actions, it can be noted that there is a underlying behavioral pattern. However, unifying them to a single value is challenging with traditional methods. Fuzzy Logic provides a means to interpret it in a more interpretable form based on its natural fluctuations.

Methodology

Step 1: Initialization

We first initialize a Mamdani Fuzzy Inference System (FIS). Such a system allows us to generate a final fuzzy output which can then be utilized in the next phase of the LSTM network.

Step 2: Membership Functions (MF)

We define membership functions to map the raw PSD values to new outputs based on the MF. A Gaussian MF was selected with PSD values in the range of 0 – 100 with an interval of 10 for values from 0 – 60 and with an interval of 20 beyond that. This was applied to both the Alpha and Beta PSD Bands. The MF of Alpha PSD is shown in Figure 5.16.

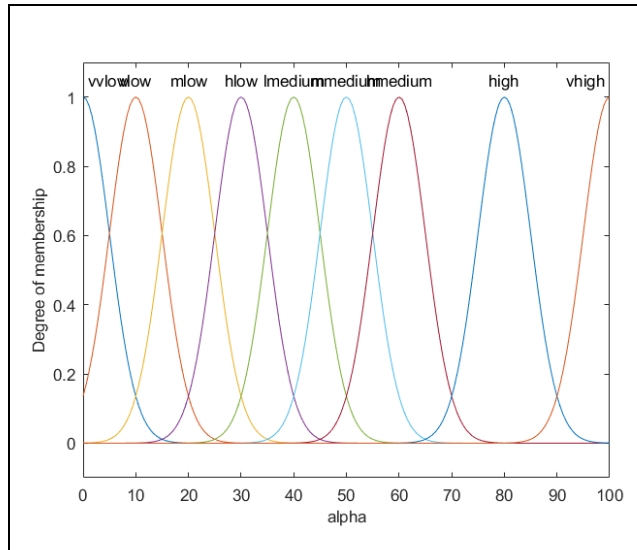


Fig 5.16: MF of Alpha PSD

Step 3: Specifying Rule Base

In this step, we specify the rules that will fuse the Alpha and Beta PSD values to generate a single psd_score.

Table 5.6: Rule Base for Alpha and Beta PSD values

Rule #1	<i>"If alpha is vlow and beta is vlow then psd_score is vlow"</i>
Rule #2	<i>"If alpha is vlow and beta is vlow then psd_score is vlow"</i>
Rule #3	<i>"If alpha is lmedium and beta is lmedium then psd_score is lmedium"</i>
Rule #4	<i>"If alpha is mmedium and beta is mmedium then psd_score is mmedium"</i>
Rule #5	<i>"If alpha is hmedium and beta is hmedium then psd_score is hmedium"</i>
Rule #6	<i>"If alpha is high and beta is high then psd_score is high"</i>
Rule #7	<i>"If alpha is vhigh and beta is vhigh then psd_score is vhigh"</i>
Rule #8	<i>"If alpha is high and beta is vlow then psd_score is mmedium"</i>
Rule #9	<i>"If alpha is high and beta is vlow then psd_score is lmedium"</i>
Rule #10	<i>"If alpha is vhigh and beta is vlow then psd_score is hmedium"</i>
Rule #11	<i>"If alpha is vhigh and beta is vlow then psd_score is mmedium"</i>
Rule #12	<i>"If alpha is vlow and beta is high then psd_score is hlow"</i>
Rule #13	<i>"If alpha is vlow and beta is vhigh then psd_score is mmedium"</i>

Rule #14 "If alpha is vlow and beta is high then psd_score is lmedium"

Rule #15 "If alpha is vlow and beta is vhigh then psd_score is hmedium"

Step 4: Completing the FIS

The FIS model is complete when all the blocks are interconnected. The surface plot gives a pictorial view of the interpretation of the rules. The complete FIS model and its surface plot can be seen in Figure 5.17.

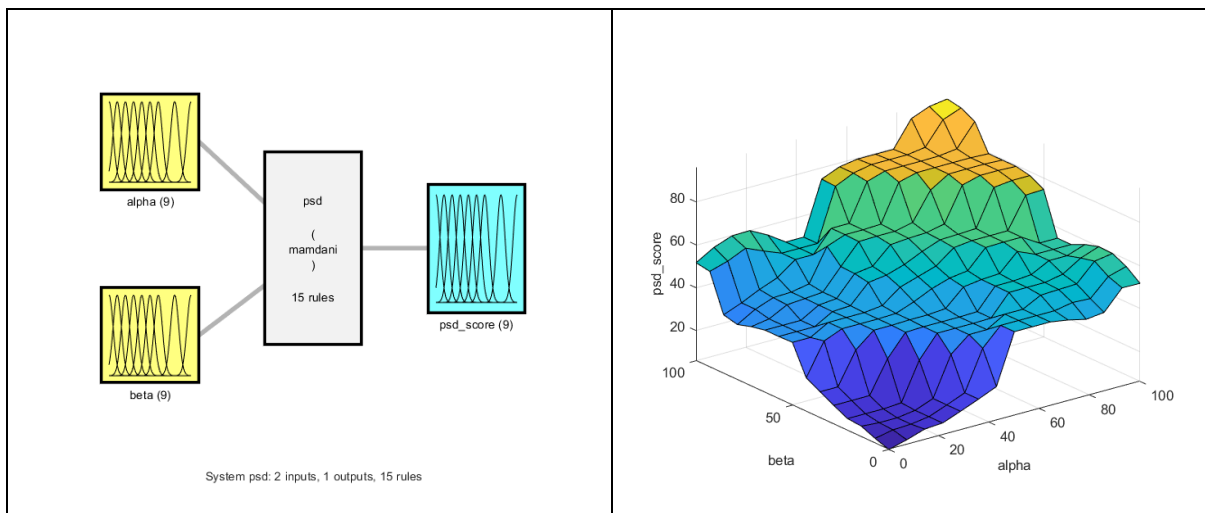
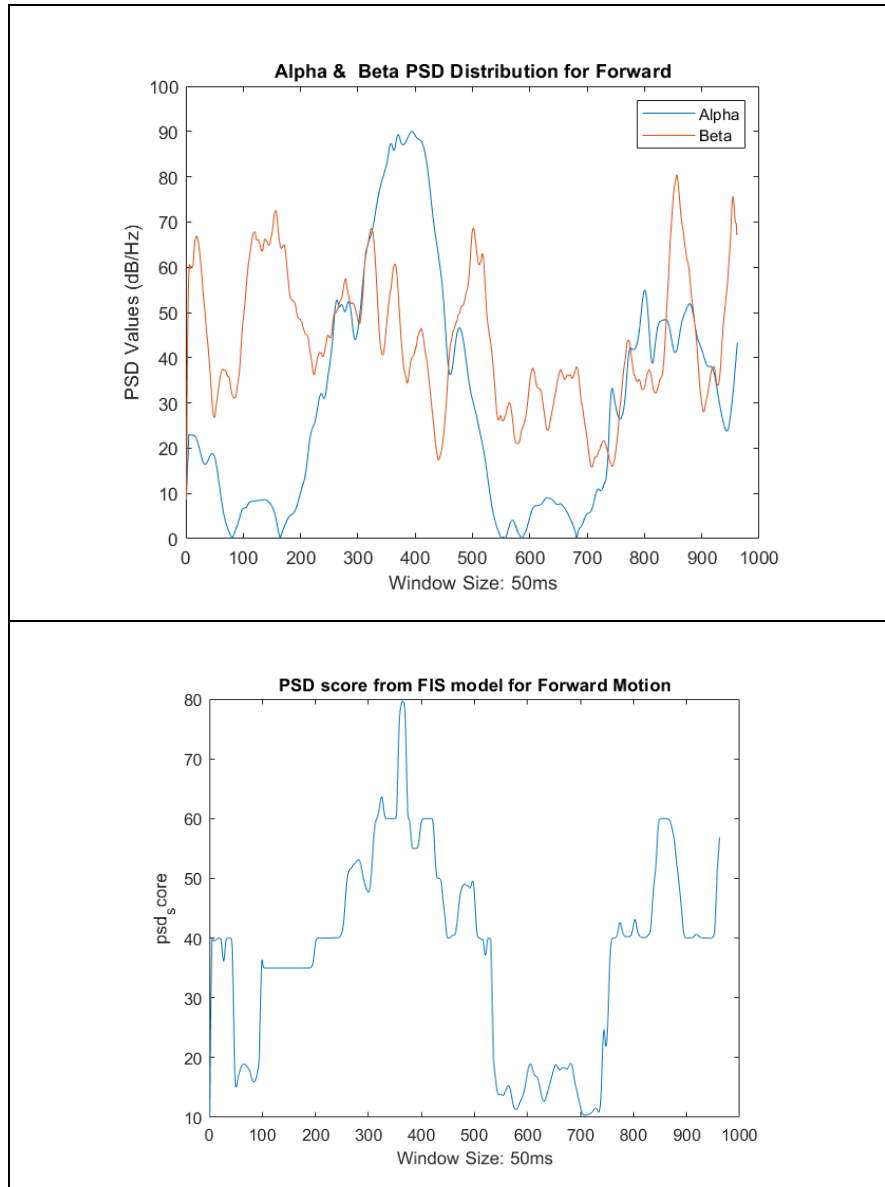


Figure 5.17: Complete FIS Model and Surface Plot

Testing and Results

The raw Alpha and Beta PSD values are fed into the FIS model and the output is generated as shown below in Figure 5.18.



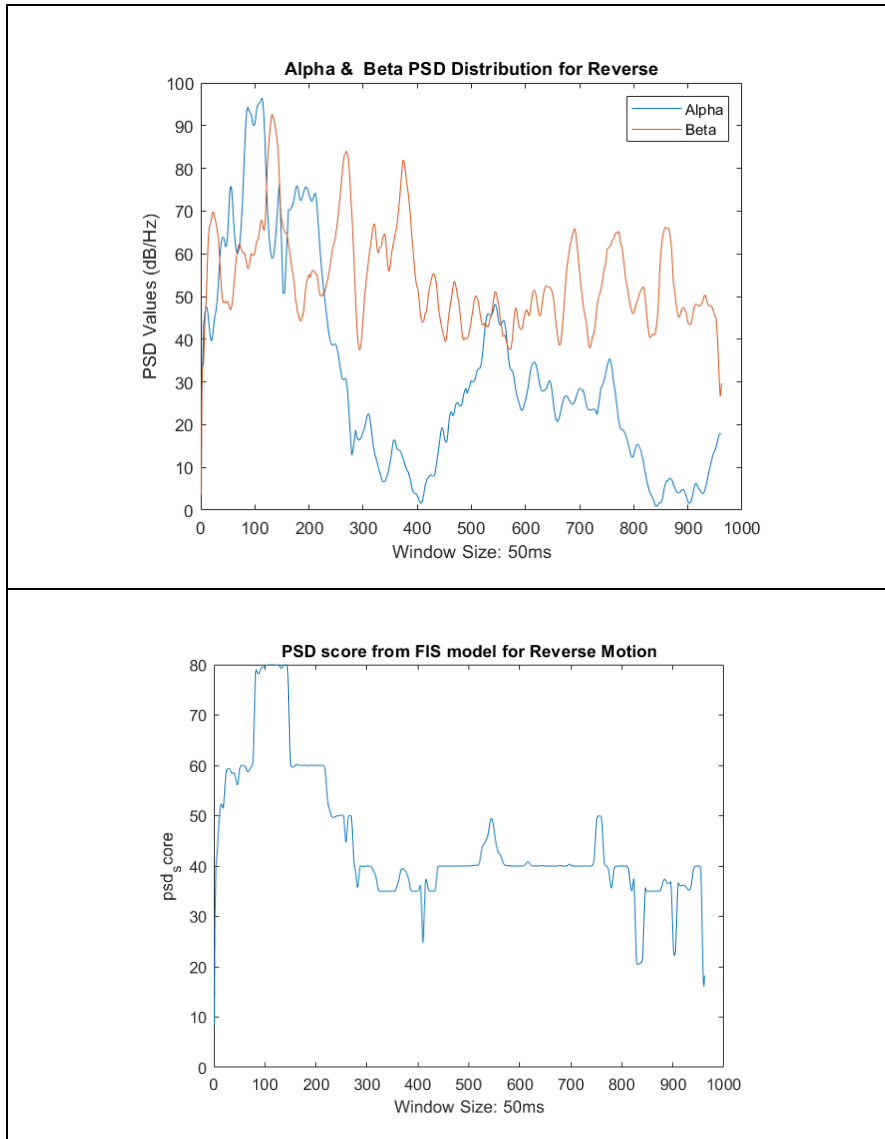


Fig 5.18: Alpha and Beta Raw PSD Values and their corresponding PSD Score from the FIS Model

The PSD Score is then used as the input for the LSTM model and the results are as shown in Figure 5.19.

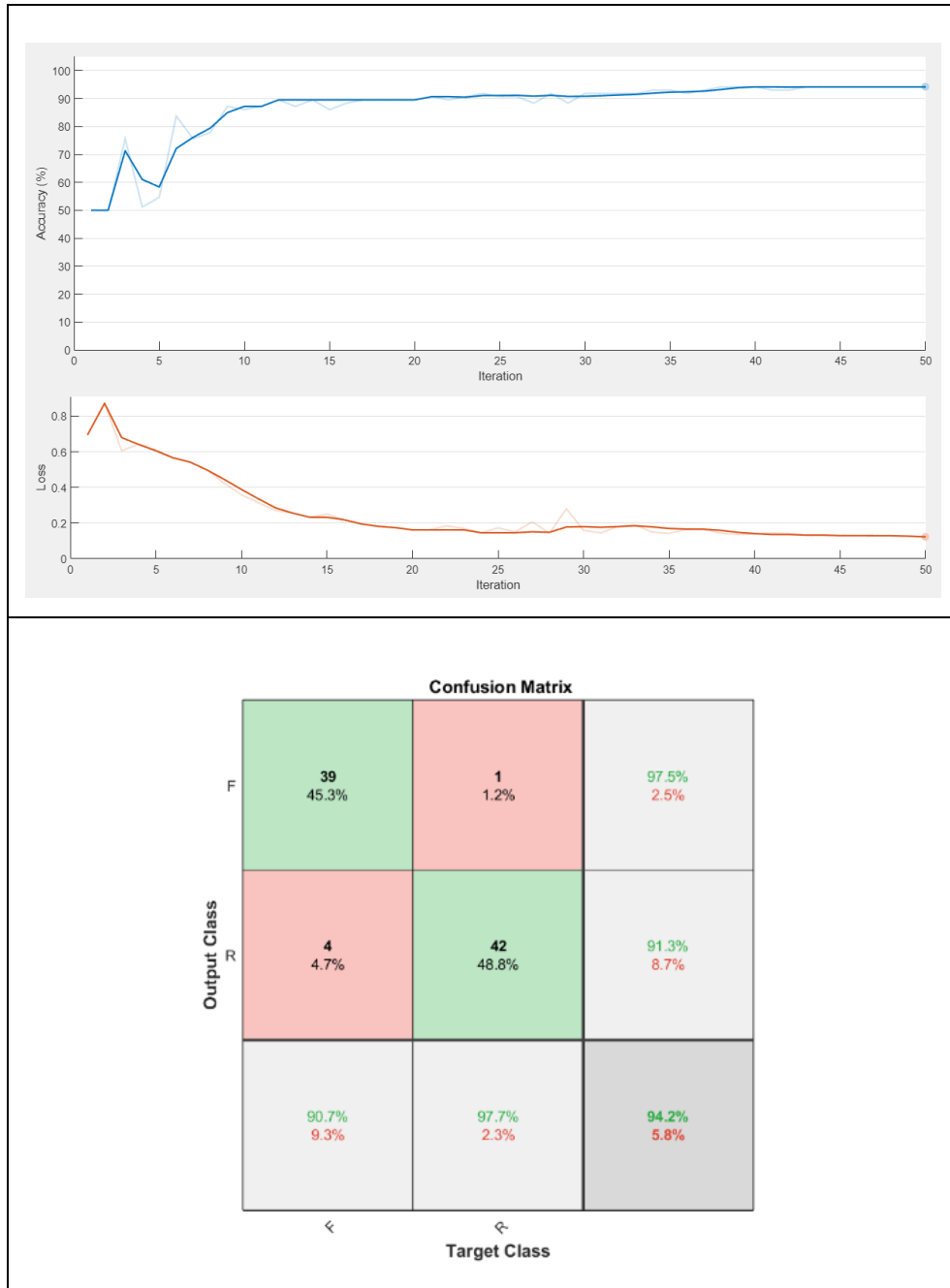


Fig 5.19: Results for FIS+LSTM Model (Forward/Reverse Action)

We are able to achieve a classification accuracy of up to **94.2%** for the Forward/Reverse Action. This is an approximately 28% improvement from using the LSTM with Time-Series Data and 20% improvement from using LSTM raw PSD Data.

5.5 EMG ANALYSIS

EMG signals exhibit both time and frequency changes in the specific sensors that are associated with the type of activity or stimulus. To fully understand the changes in these signals, we need to be able to analyze the signal over a subset of time. This requires a solution that has the ability to remember and detect specific patterns in the signals over time. The Long Short-Term Memory (LSTM) is a network that is able to achieve this objective. It has been utilized in the analysis of various time-series data which are representative of many biological signals [211, 212]. LSTM has also been utilized in many other applications that require memory and the “context” of the data with respect to its past. Natural Language Processing (NLP) is such an example.

Sensor Placement

The 5 EMG sensors were placed on pertinent right arm muscles. The chosen locations were the anterior first dorsal interosseus, common extensor digitorum, flexor digitorum, brachioradial and the deltoid muscles. The positions of the sensor on a participant can be seen in Figure 5.20.

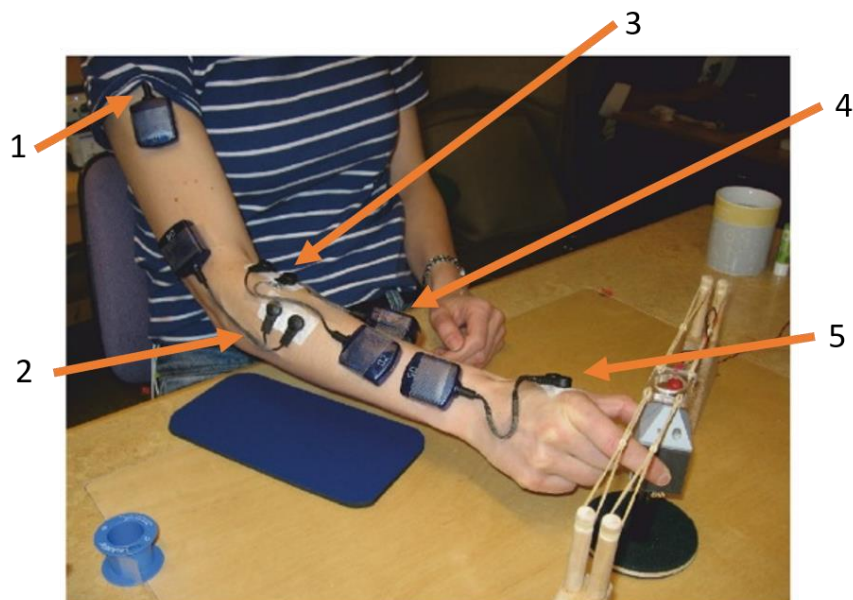


Fig 5.20: EMG sensor placement: 1-anterior deltoid, 2-brachioradialis, 3-flexor digitorum, 4-common extensor digitorum, 5-first dorsal interosseus

In each trial, the participants performed the same task of moving their arm forward to pick-up an object to a fixed height. This was cued by an LED being turned ON. When the LED is turned OFF, the participant would lower the object and return the arm back to the resting state. The sequence of the movements is shown in Fig 5.21.

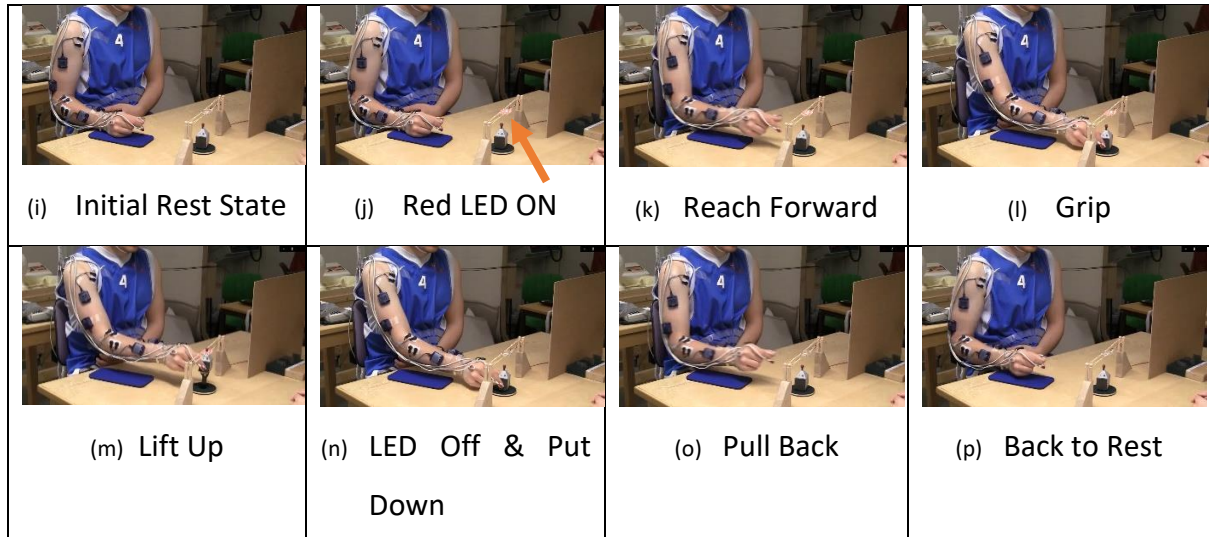
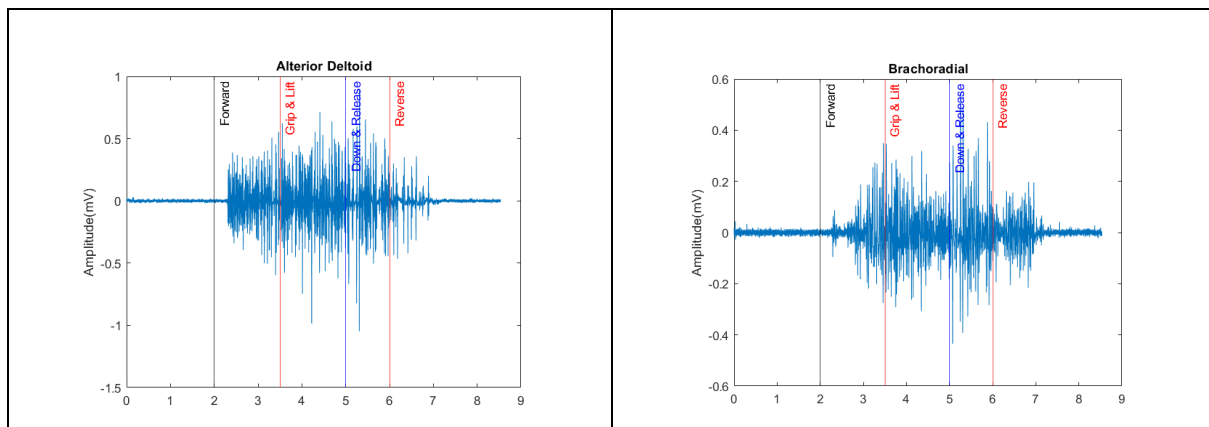


Fig 5.21: Sequence of Actions performed by Participant

The raw data captured from the EMG sensors is shown in Figure 5.22. The markers for the LEDOn, LEDOff, GripUp and RelDown are shown.



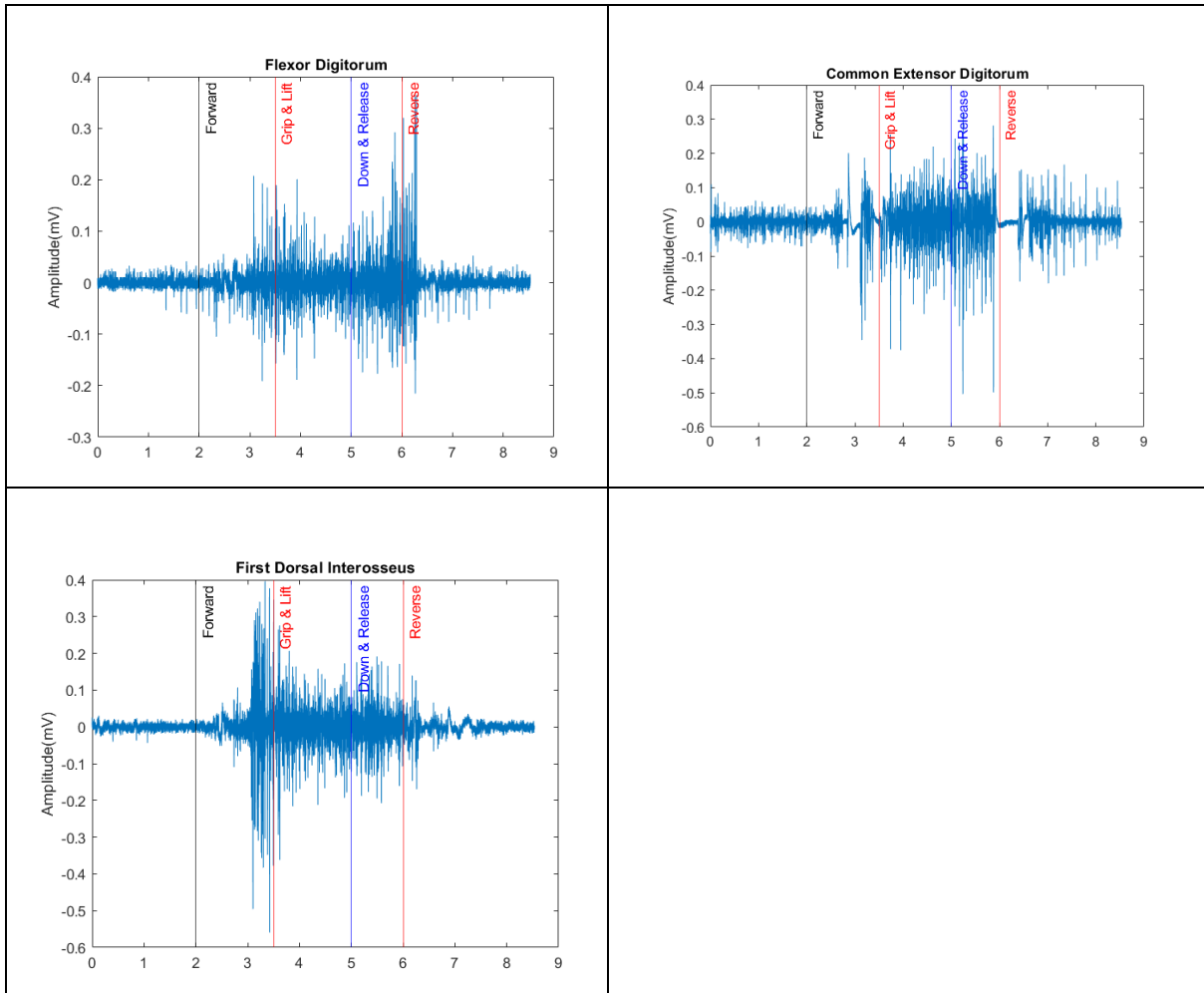
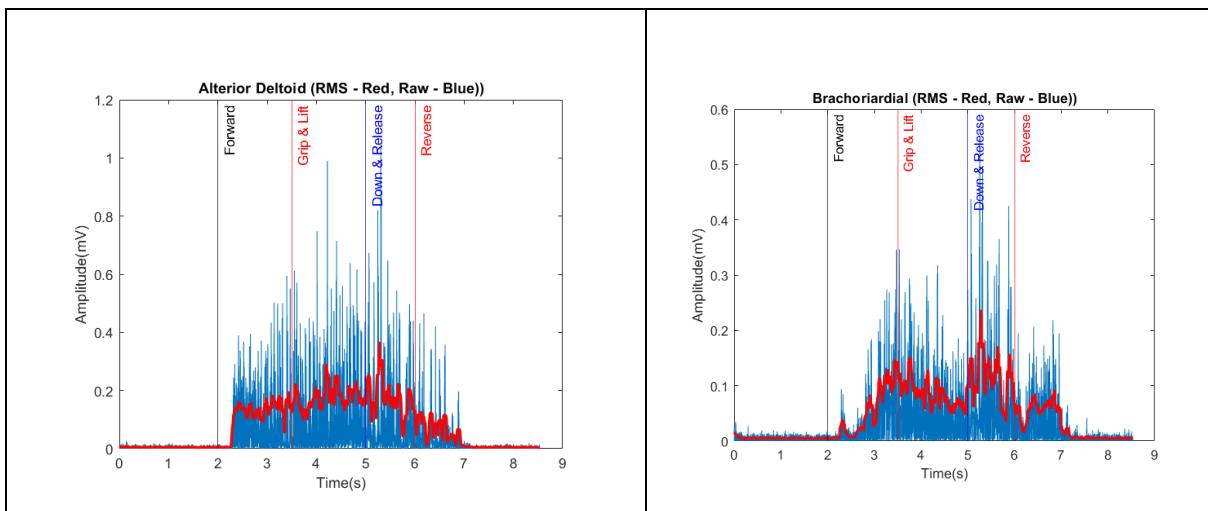


Fig 5.22: Raw EMG Data with Time Markers

The data is then filtered and the RMS is computed. The waveforms are as shown below in Figure 5.23.



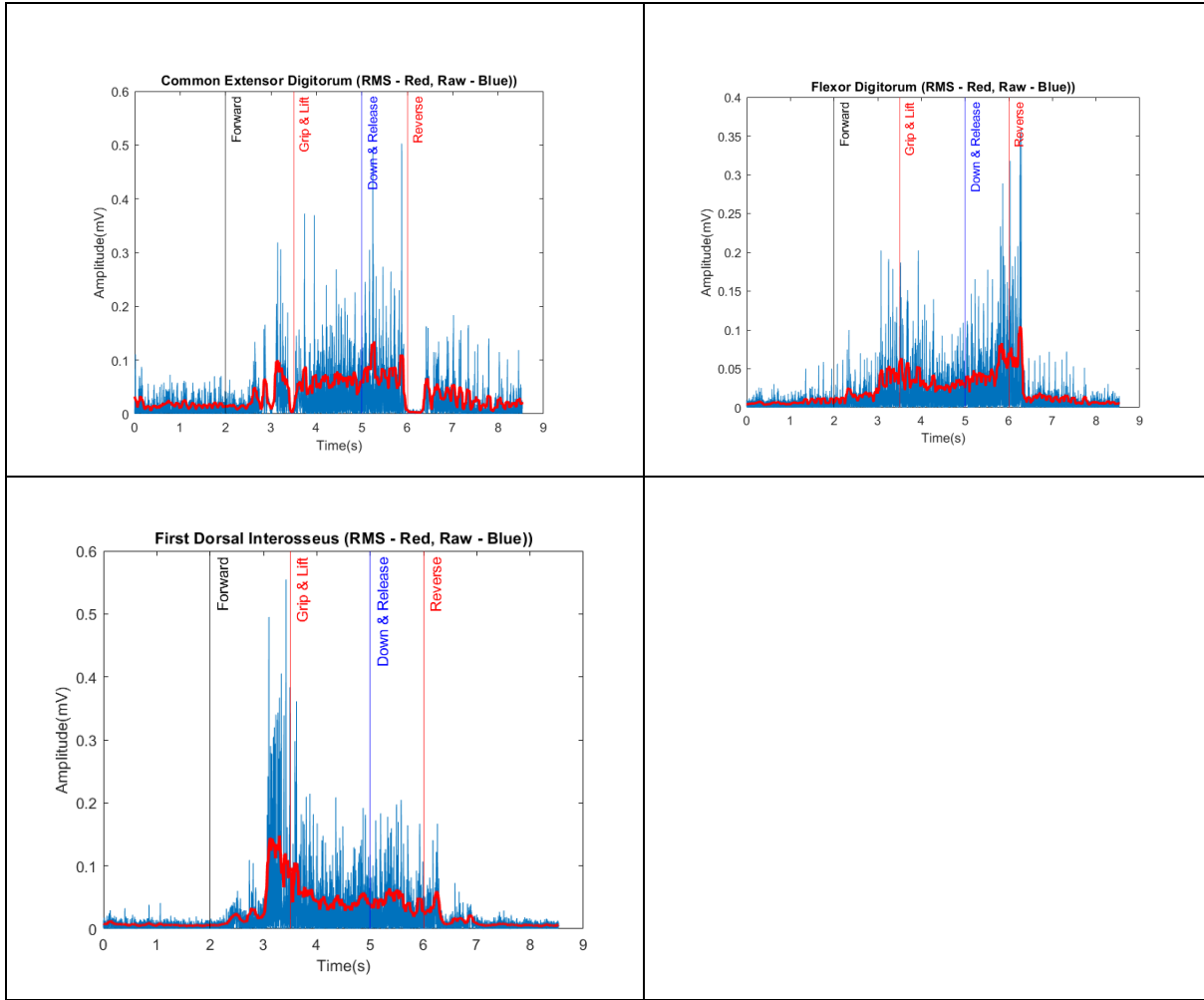


Fig 5.23: RMS Data (Red) with Raw EMG Data (Blue) with Time Markers

5.6 LSTM Training and Testing

The LSTM Network was configured to have 100 cells with two fully connected layers. The processed data was split into 70% Training and 30% Testing sets. Both training and testing was done for 50 epochs with 10-fold cross validation being performed. Accuracy, Recall, Precision and F1-Score were computed based on the True Positive (TP), True negative (TN), False Positive (FP) and False Negative (FN) values.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad \text{Eq. 5.14}$$

$$F1 - score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad \text{Eq. 5.15}$$

In the first stage, the raw RMS values are used to train and test the LSTM network.

Forward – Reverse Analysis for AD channel

Figure 5.24 shows the Training and Testing Phase for the LSTM Network.



Fig 5.24: LSTM performance for Training (a) and Testing (b) Phase

The confusion matrix for the training and testing phases are shown in Figure 5.25.



Fig 5.25: Confusion Matrix for Training and Testing Phase

As can be seen, the classification accuracy based on RMA time domain data is very low, in the range of 48.5 – 50.0%. The RMS time-series analysis was carried out for the remaining EMG sensors. The results are as shown in Table 5.7.

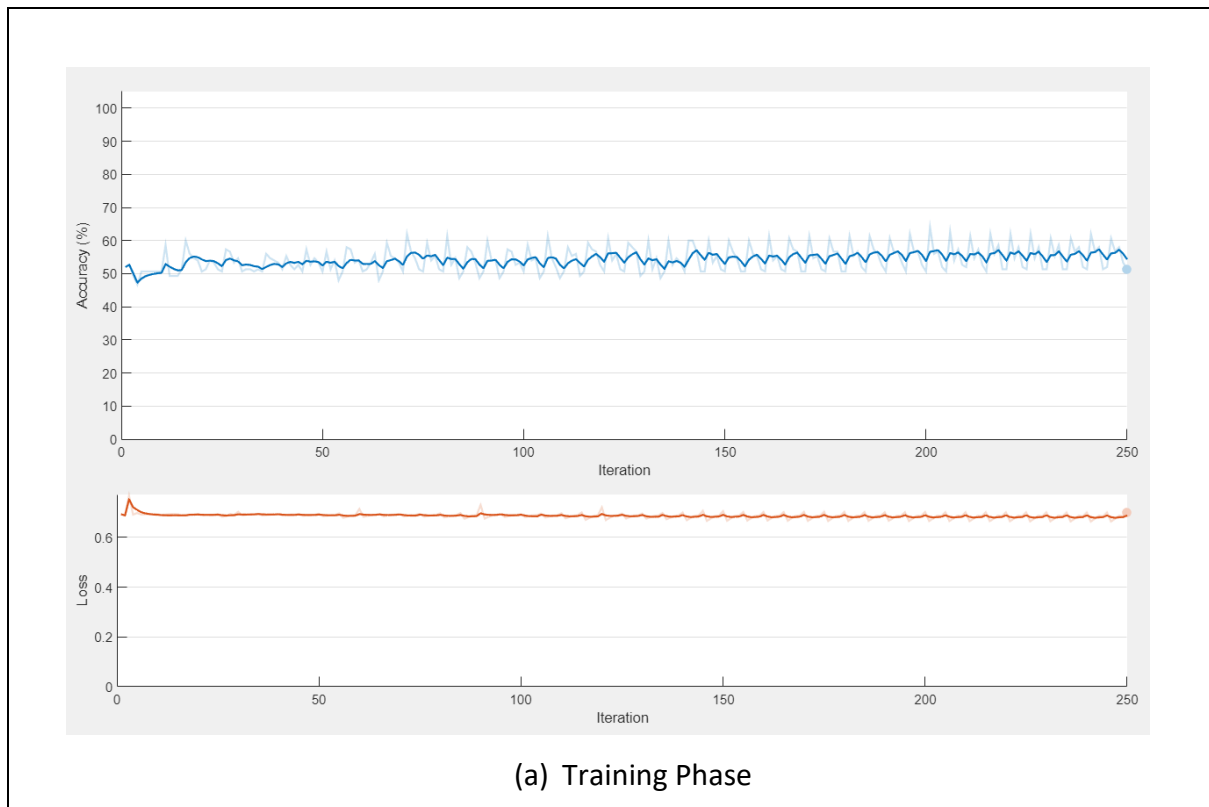
Table 5.7: Forward/Reverse Classification Accuracy for RMS Time-Series Data

Channel	Accuracy Range	F1-Score
AD	48.5% - 50.0%	0.4855
BR	47.2% - 49.3%	0.4723
FD	44.7% - 50.5%	0.4632
CED	45.6% - 48.2%	0.4511
FDI	43.3% - 47.5%	0.4387

As can be observed, using a single channel's EMG RMS time-series is not very effective to provide a high level of classification accuracy for the Forward / Reverse motion.

Grip Up – Release Down Analysis

The methodology is repeated for the Grip Up / Release Down analysis. Figure 5.26 shows the Training and Testing Phase for the LSTM Network.



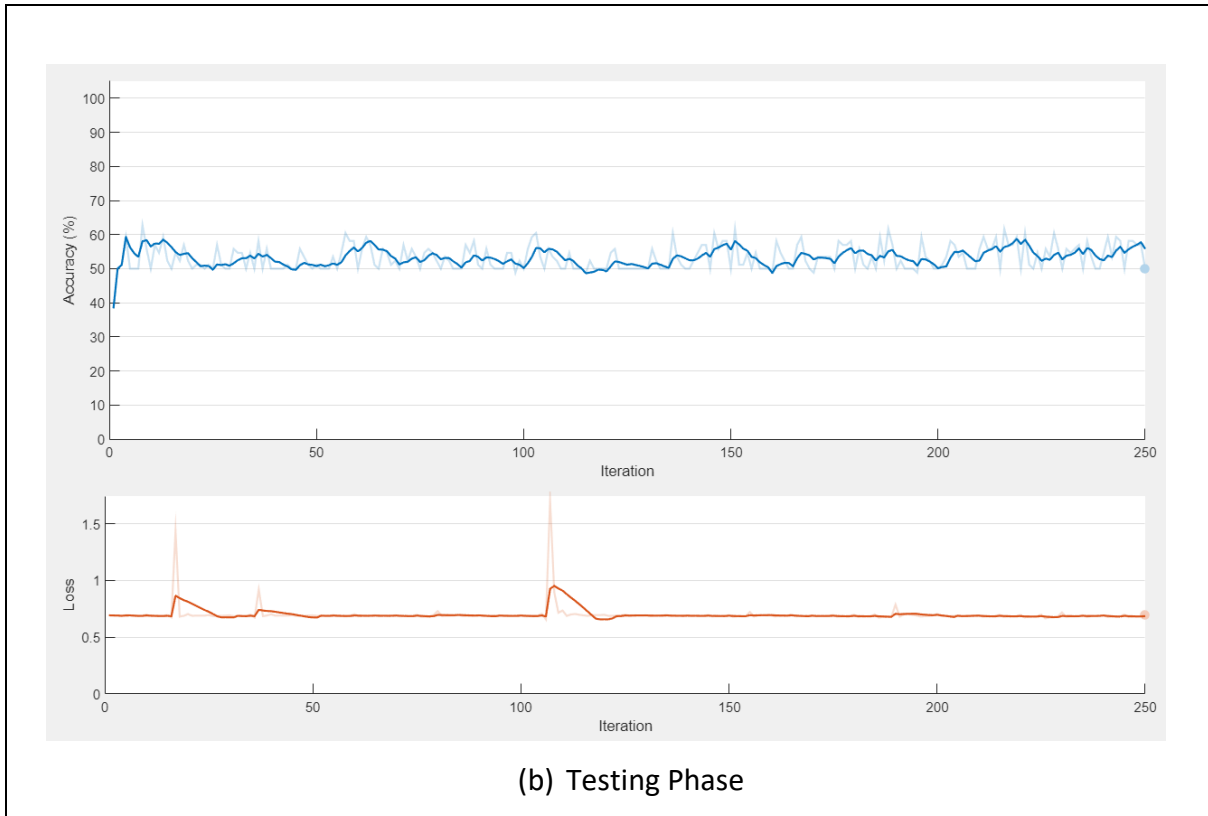
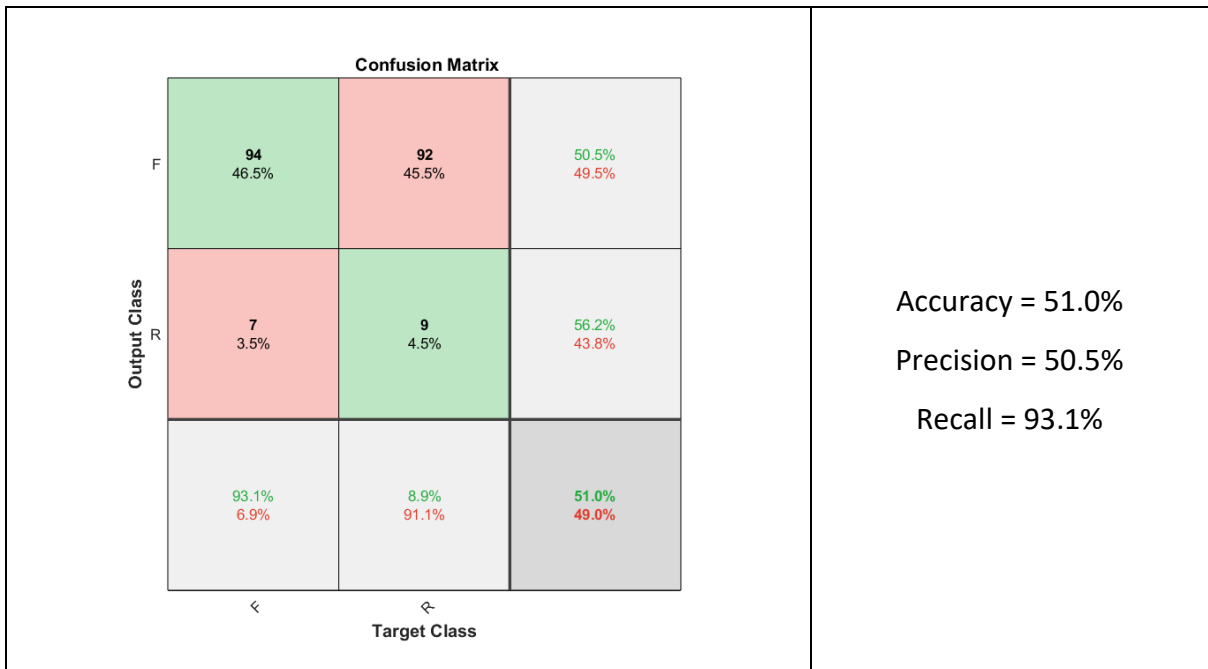


Fig 5.26: LSTM performance for (a) Training and (b) Testing Phase

The confusion matrix for the training and testing phases are shown in Figure 5.27.



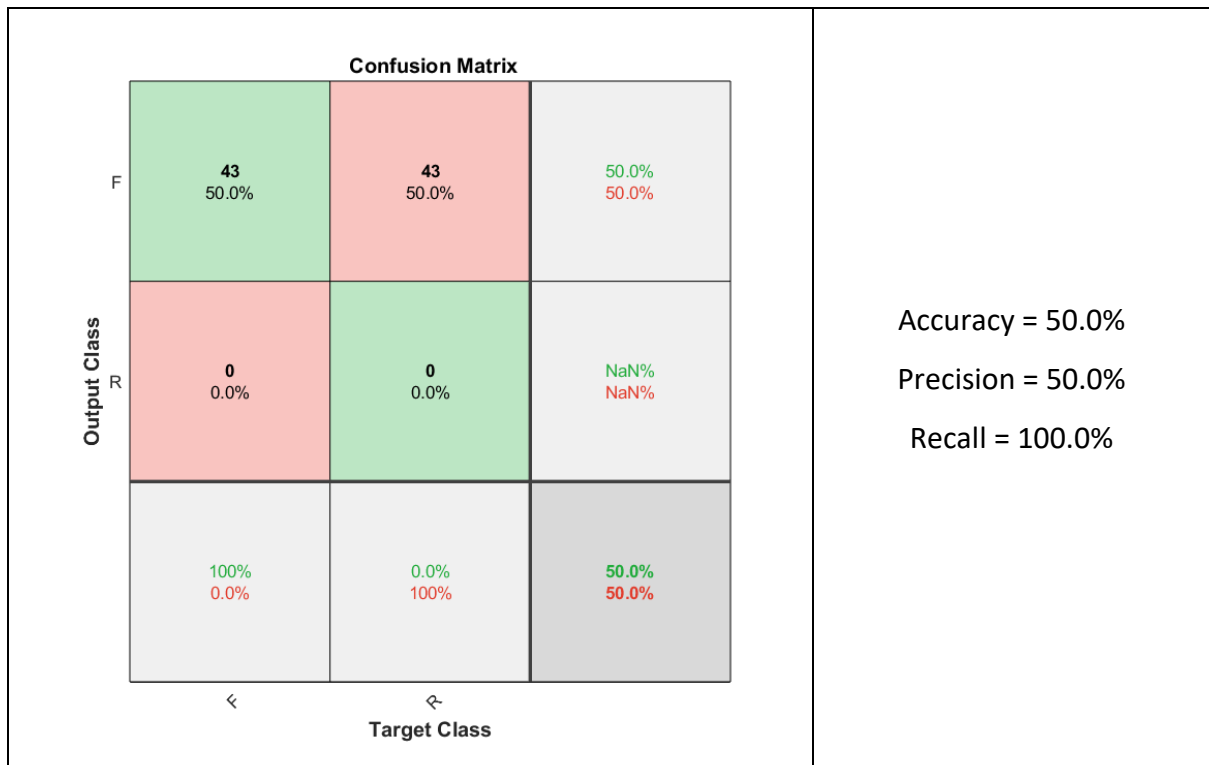


Fig 5.27: Confusion Matrix for Training and Testing Phase

The RMS time-series analysis was carried out for the remaining EMG sensors. The results are as shown in Table 5.8.

Table 5.8: GripUp / RelDown Classification Accuracy for RMS Time-Series Data

Channel	Accuracy Range	F1-Score
AD	50.0% - 51.0%	0.4921
BR	49.1% - 50.2%	0.4865
FD	45.2% - 49.5%	0.4521
CED	44.2% - 49.1%	0.4344
FDI	42.9% - 46.4%	0.4289

As can be observed, using a single channel's EMG RMS time-series is not very effective to provide a high level of classification accuracy for the GripUp / RelDown motion.

5.7 Applying Fuzzy Logic to EMG Signals

Observation of the RMS Signals for the Forward and Reverse Actions, it can be noted that there is a underlying behavioral pattern. However, unifying them to a single value is challenging with traditional methods. Fuzzy Logic provides a means to interpret it in a more interpretable form based on its natural fluctuations. We first analyze the EMG signals for all 5 sensors for the different windows. Figure 5.28 shows the Forward movement window for all EMG channels. It can be observed from the figure, that the range of the values of each of the sensors is distinct. In order to fuse these signals together in the FIS, the range has to be uniform. The first step is to perform a normalization and keep all sensor values in the range of 0 to 1.

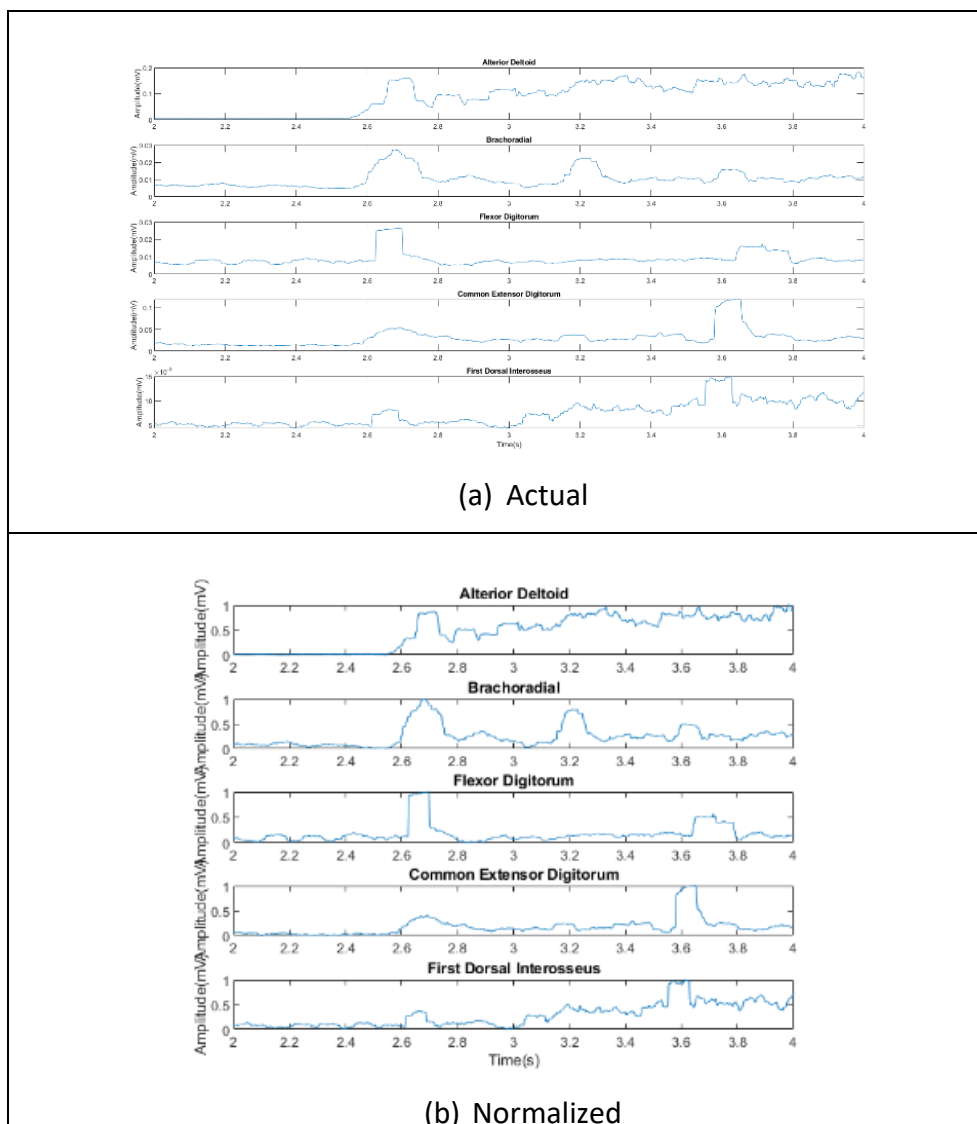


Figure 5.28: EMG RMS values for all 5 sensors

With these normalized values, we can now proceed with the creation of the FIS.

Methodology

Step 1: Initialization

We first initialize a Mamdani Fuzzy Inference System (FIS). Such a system allows us to generate a final fuzzy output which can then be utilized in the next phase of the LSTM network.

Step 2: Membership Functions (MF)

We define membership functions to map the raw RMS values to new outputs based on the MF. As shown in Figure 5.29, a Gaussian MF was selected with RMS values in the range of 0 – 1 with an interval of 0.2 for all values. This was applied to both the Forward and Reverse movements across all five channels.

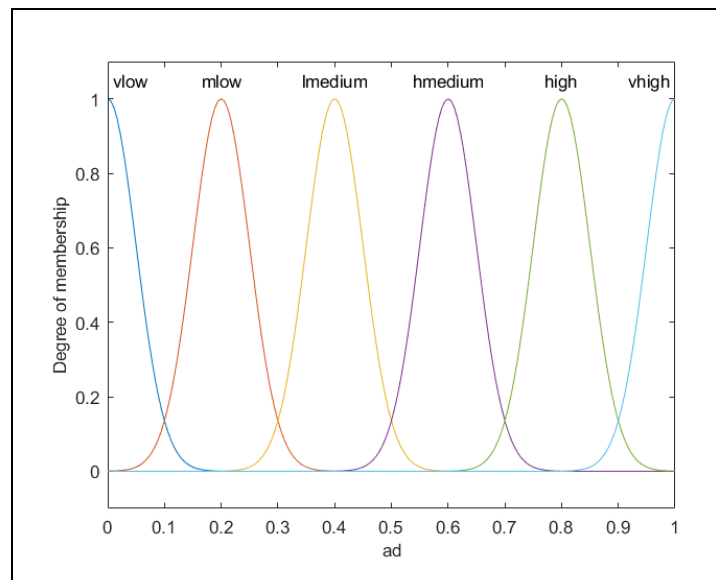


Fig 5.29: MF of AD Sensor

Step 3: Specifying Rule Base

Table 5.9 specifies the rules that will fuse the AD, FD and BR values to generate a single emg_score.

Table 5.9: Rule Base for AD, BR and FD RMS values (Forward / Reverse)

Rule #1	<i>"If ad is vlow and br is vlow then emg_score is vlow"</i>
Rule #2	<i>"If ad is mlow and br is mlow then emg_score is lmedium";</i>
Rule #3	<i>"If ad is hmedium and br is hmedium then emg_score is hmedium";</i>

-
- Rule #4** "If fd is vlow and br is vlow then emg_score is vlow";
-
- Rule #5** "If fd is mlow and br is mlow then emg_score is lmedium";
-
- Rule #6** "If fd is hmedium and br is hmedium then emg_score is hmedium";
-
- Rule #7** "If ad is high and br is high then emg_score is high";
-
- Rule #8** "If ad is vhigh and br is vhigh then emg_score is vhigh";
-
- Rule #9** "If ad is high and br is vlow then emg_score is mmedium";
-
- Rule #10** "If fd is high and br is high then emg_score is high";
-
- Rule #11** "If fd is vhigh and br is vhigh then emg_score is vhigh";
-
- Rule #12** "If fd is high and br is vlow then emg_score is mmedium";
-
- Rule #13** "If ad is vhigh and br is vlow then emg_score is hmedium";
-
- Rule #14** "If ad is vlow and br is high then emg_score is lmedium";
-
- Rule #15** "If ad is vlow and br is vhigh then emg_score is hmedium";
-

Step 4: Completing the FIS

The FIS model is complete when all the blocks are interconnected. The surface plot gives a pictorial view of the interpretation of the rules. The complete FIS model and its surface plot is shown in Figure 5.30. For the interpretation of the Forward and Reverse movements, only the AD, BR and FD channels are used.

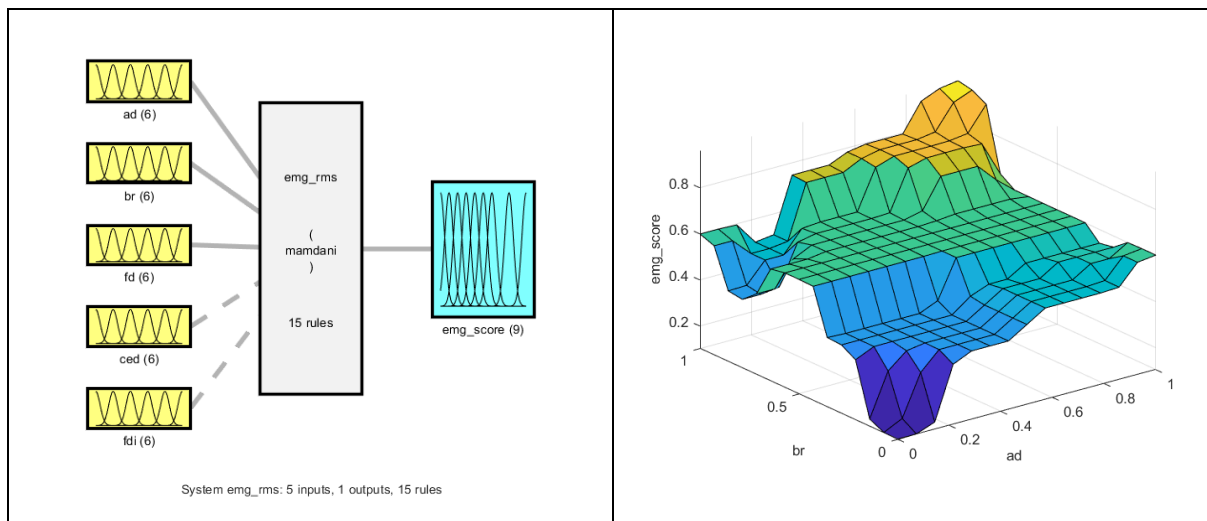


Figure 5.30: Complete FIS Model and Surface Plot

Testing and Results

The raw AD, BR and FD RMS values are fed into the FIS model and the output is generated. Figure 5.31 shows the RMS values (left column) and the generated RMS scores (right column).

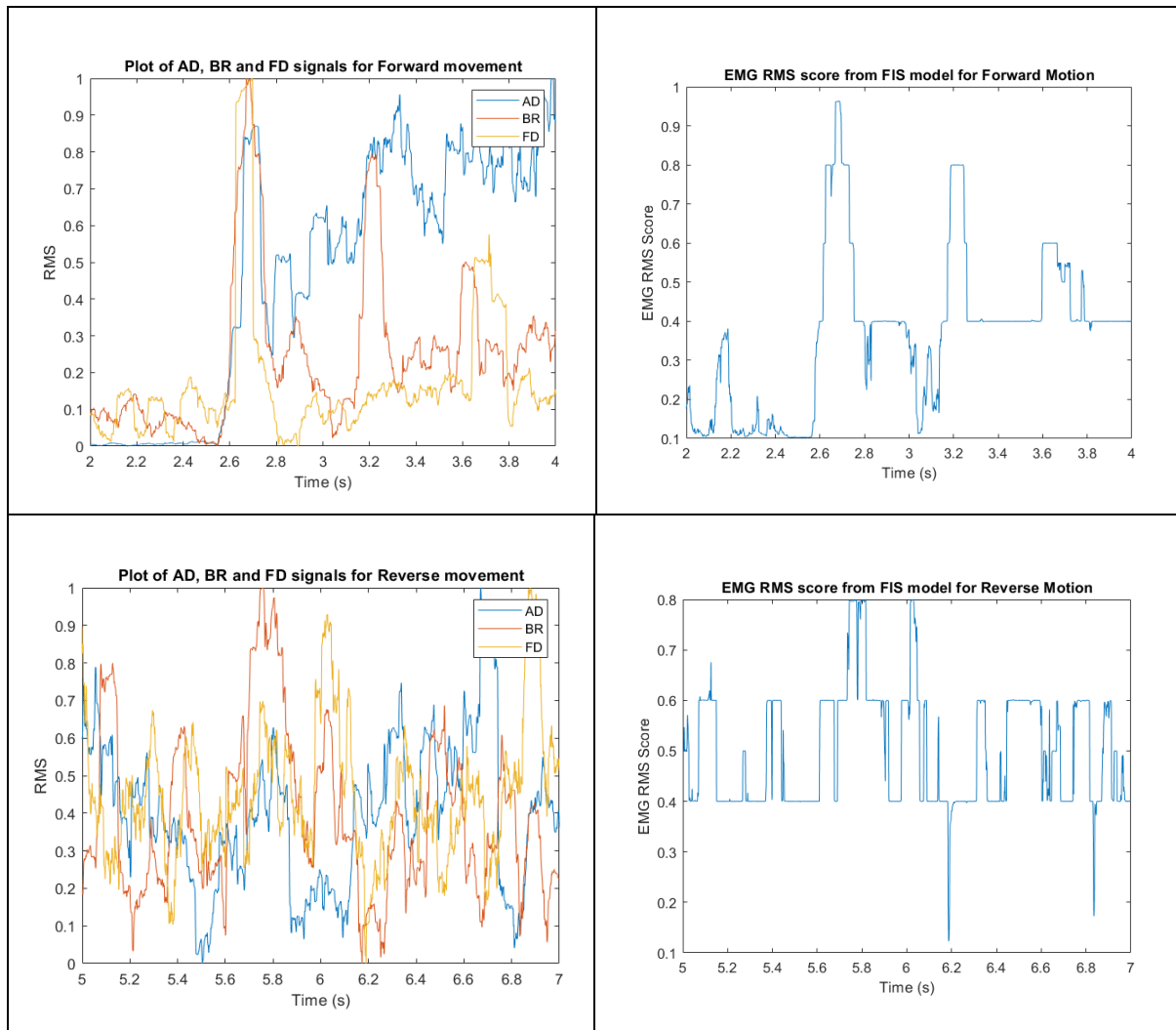


Fig 5.31: AD, BR and FD channel RMS values and their corresponding EMG Score from the FIS Model

The EMG Score is then used as the input for the LSTM model and the results are computed. We are able to achieve a classification accuracy of up to **95.1%** for the Forward/Reverse Action. This is an approximately 45% improvement from using the LSTM with Time-Series RMS Data.

Analysis of GripUp & RelDown movements

A similar analysis is done for the GripUp and RelDown movements. For these actions, the FD, CED and FDI channels are more prominent and are used in the FIS rules, as shown in Table 5.10.

Table 5.10: Rule Base for AD, BR and FD RMS values (GripUp / RelDown)

Rule #1	<i>"If ced is vlow and fdi is vlow then emg_score is vlow"</i>
Rule #2	<i>"If ced is mlow and fdi is mlow then emg_score is lmedium";</i>
Rule #3	<i>"If ced is hmedium and fdi is hmedium then emg_score is hmedium";</i>
Rule #4	<i>"If fd is vlow and fdi is vlow then emg_score is vlow";</i>
Rule #5	<i>"If fd is mlow and fdi is mlow then emg_score is lmedium";</i>
Rule #6	<i>"If fd is hmedium and fdi is hmedium then emg_score is hmedium";</i>
Rule #7	<i>"If ced is high and fdi is high then emg_score is high";</i>
Rule #8	<i>"If ced is vhigh and fdi is vhigh then emg_score is vhigh";</i>
Rule #9	<i>"If ced is high and fdi is vlow then emg_score is mmedium";</i>
Rule #10	<i>"If fd is high and fdi is high then emg_score is high";</i>
Rule #11	<i>"If fd is vhigh and fdi is vhigh then emg_score is vhigh";</i>
Rule #12	<i>"If fd is high and fdi is vlow then emg_score is mmedium";</i>
Rule #13	<i>"If ced is vhigh and fdi is vlow then emg_score is hmedium";</i>
Rule #14	<i>"If ced is vlow and fdi is high then emg_score is lmedium";</i>
Rule #15	<i>"If ced is vlow and fdi is vhigh then emg_score is hmedium";</i>

The FIS model and its surface plot are as shown in Figure 5.32.

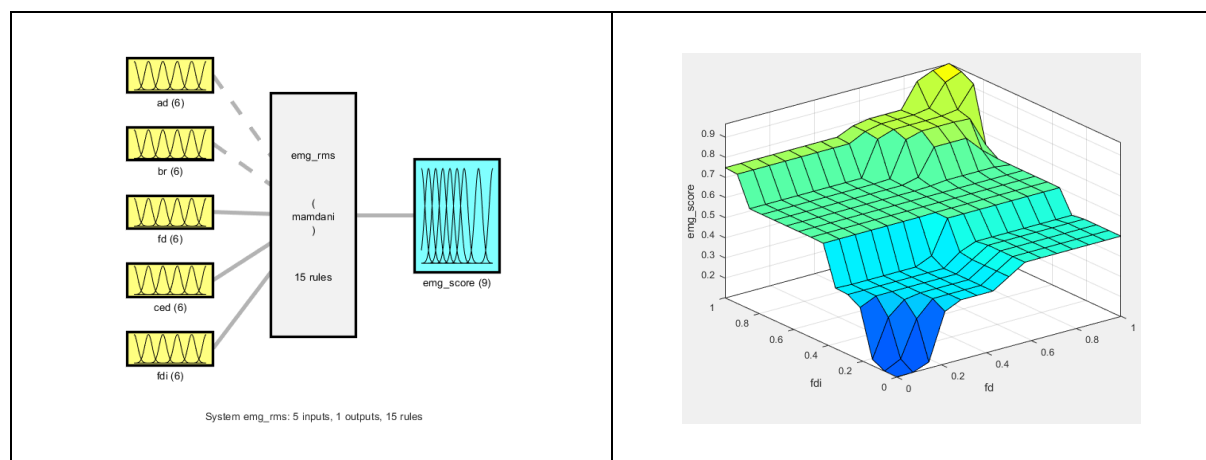


Figure 5.32: FIS Model and Surface Plot for FD & FDI channels (GripUP / RelDown)

In Figure 5.33, we can see the raw EMG signals and the combined emg_score obtained through the FIS.

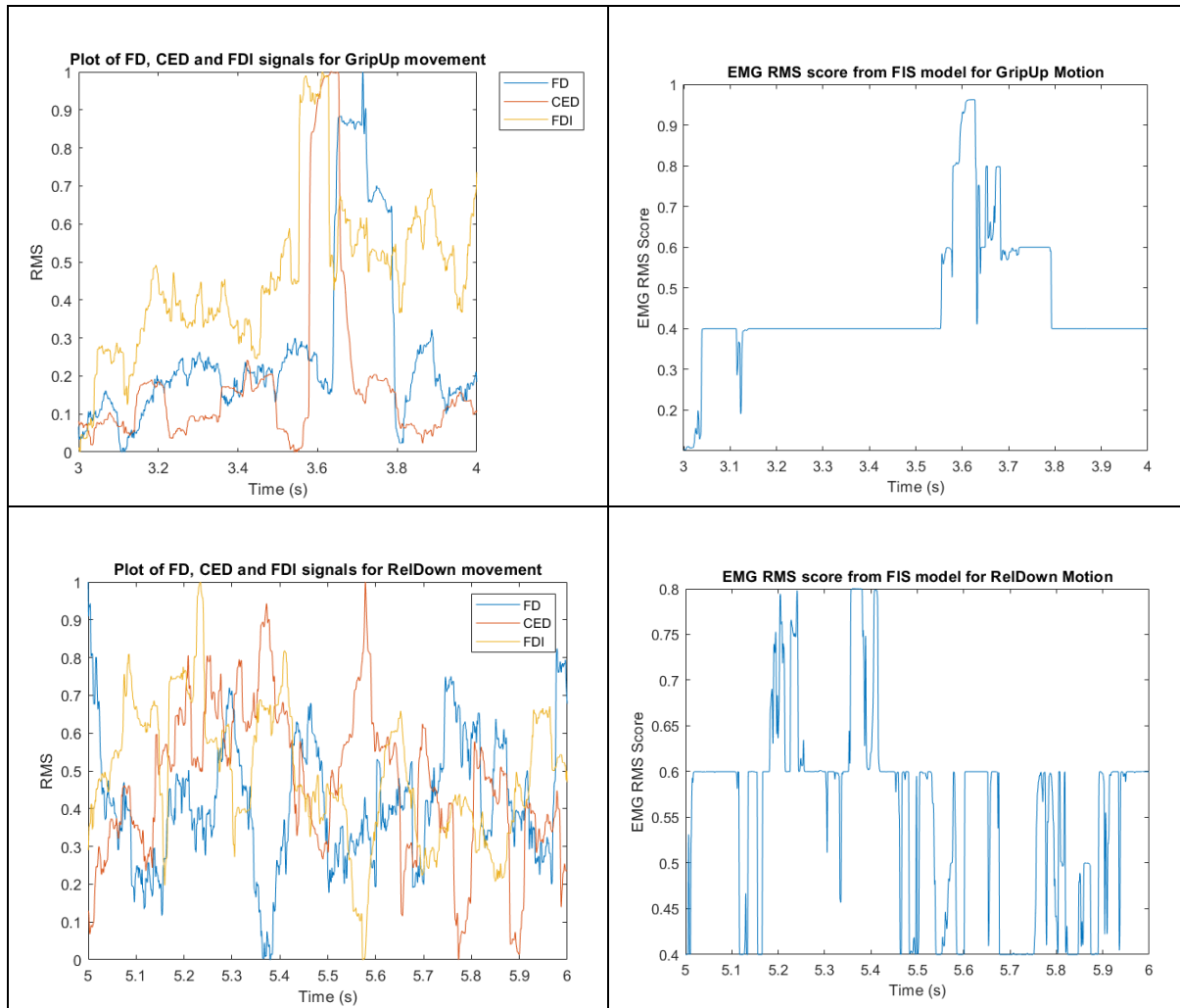


Figure 5.33: FD, CED and FDI channel RMS values and their corresponding EMG Score from the FIS Model

The EMG Score is then used as the input for the LSTM model and the results are computed. We are able to achieve a classification accuracy of up to **96.7%** for the GripUp/Release Action. This is an approximately 47% improvement from using the LSTM with Time-Series RMS Data.

A comparative study was performed to evaluate the effectiveness of LSTM against other Deep Learning Techniques such as Convolutional Neural Networks (CNN) and Deep-Belief Networks (DBN). For the proposed Long Short-Term Memory (LSTM) network, the average accuracy was

95.9% with an F1 score of 0.9467. For the DBN, the accuracy achieved was 92.3% with an F1 score of 0.9021, and for the CNN, the accuracy was 88.1% with an F1 score of 0.8213. The Receiver Operator Characteristic (ROC) [214] is a probability curve that plots the True-Positive (TP) against the False-Positive (FP). The area under the ROC, referred to as the Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between different classes. The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes. It can be seen from Figure 5.34 that the AUC for the LSTM network is greater than the CNN and DBN.

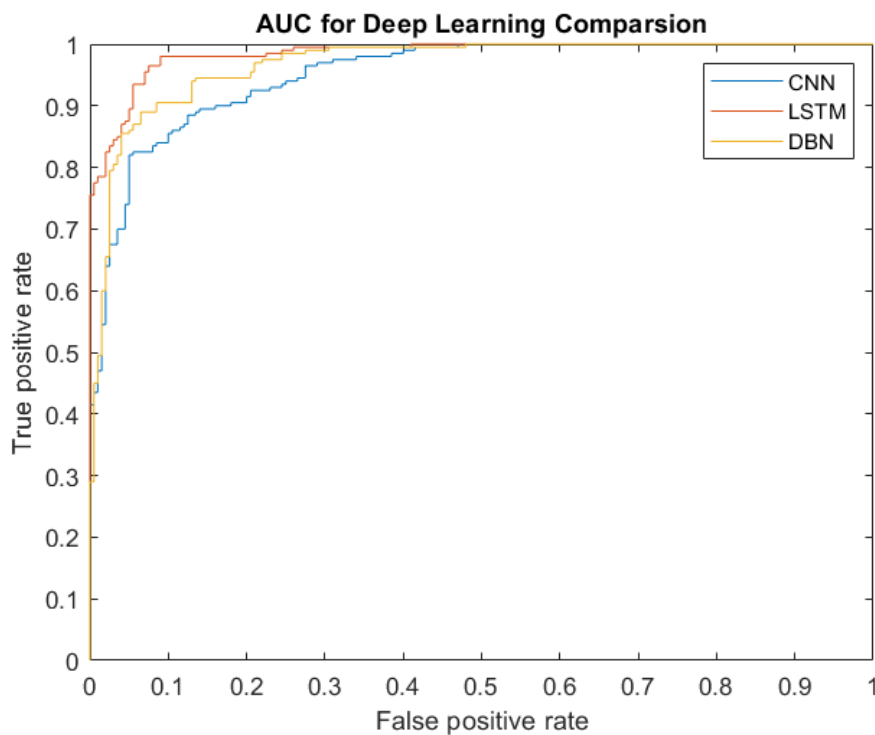


Figure 5.34: AUC for comparison of Deep Learning Methods

6.8 FOUR-WAY Classification using FIS+LSTM Model

In the final phase, 4-way classification is done with the data. Leveraging on the existing methodology, the FIS Rules are updated to incorporate all the rules for both Forward/Reverse and GripUp/RelDown action pairs.

Figure 5.35 shows the various emg_scores obtained through the combined FIS model.

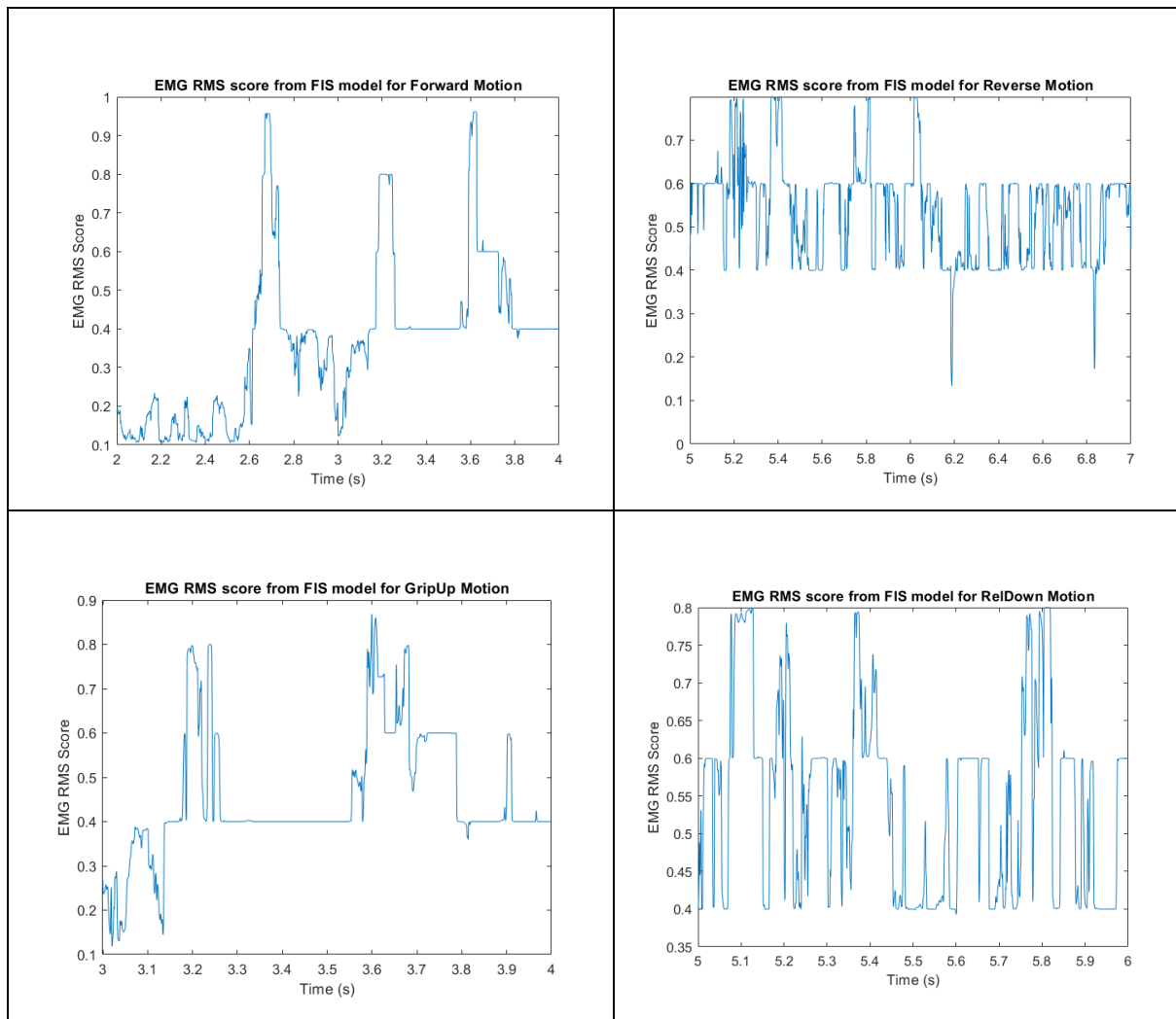


Figure 5.35: EMG scores obtained through the combined FIS model

With the combined FIS model, the overall classification accuracy for the various actions are as shown in Table 5.11 and Figure 5.36.

Table 5.11: Classification Accuracy for Various Movements

Movement	Accuracy
Forward	91.3±0.7%
Reverse	90.1±1.1%
GripUp	89.5±0.9%
RelDown	88.9±0.6%

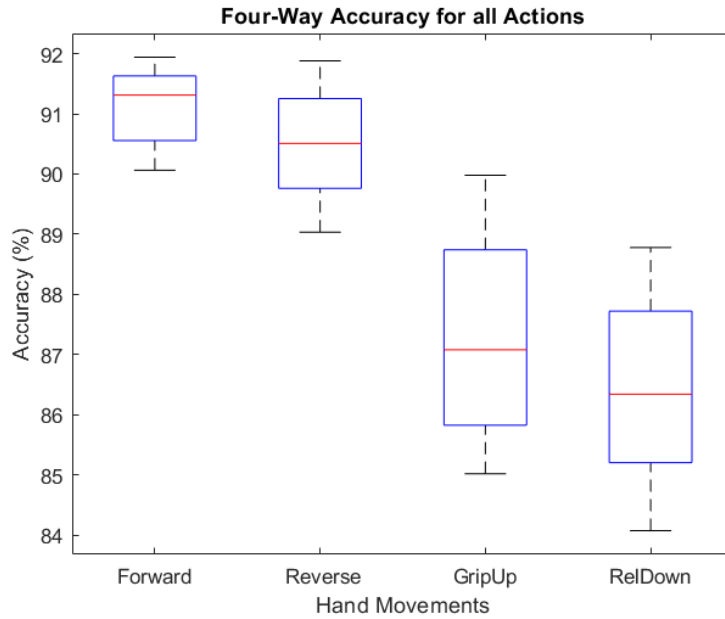


Figure 5.36: Four-Way Classification Accuracies

The associated confusion matrix is as shown in Fig 5.37.

True Class	Forward	1132	106	0	0	91.44%
	Reverse	104	1118	16	0	90.31%
	RelDown	10	12	1060	156	85.62%
	GripUp	0	0	135	1103	89.10%
Recall	90.85%	90.45%	87.53%	87.61%		
		Predicted Class				
						Precision

Fig 5.37: Four-Way Classification Accuracy

A comparison study was also performed to evaluate the proposed methodology using other deep learning networks and classifiers, such as CNN, DBN, Support Vector Machine (SVM), k-Nearest Neighbour (kNN), and the Artificial Neural Network (ANN). With the same features

and Fuzzy Rule Base, the following networks were tested: Fuzzy+CNN, Fuzzy+DBN, Fuzzy+SVM, Fuzzy+kNN, and Fuzzy+ANN. The classification accuracy results are shown in Figure 5.38. In Table 5.12 we can observe the critical network parameters together with the average four-class classification accuracies for them.

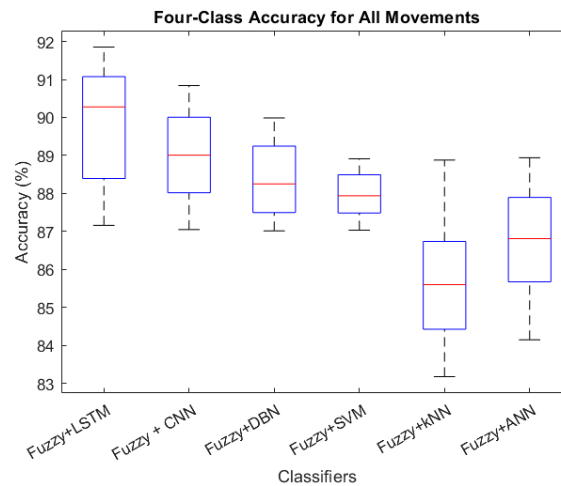


Figure 5.38: Four-Class Accuracy Comparison

The classification results are shown in Table 5.12.

Table 5.12: Accuracy and F1-Score

Classifier	Accuracy	F1-Score
Fuzzy+LSTM Parameters: HiddenUnits: 100, StateActivation: tanh, GateActivation: sigmoid	90.4±2.4%	0.8992
Fuzzy+CNN Parameters: ConvolutionalLayers: 3, maxPoolingLayers: 3, Solver: Stochastic Gradient Descent with Momentum (SGDM)	88.9±1.3%	0.8805
Fuzzy+DBN Parameters: ActivationFunction: tanh, Momentum: 0.5, Output: Sigmoid	88.1±2.3%	0.8785
Fuzzy+SVM Parameters: Kernel Function: Linear, Solver: Iterative Single Data Algorithm (ISDA)	87.5±1.6%	0.8621

Fuzzy+kNN Parameters: Distance: Euclidean, NumNeighbours: 6	85.8±2.9%	0.8454
Fuzzy+ANN Parameters: Hidden Layer: 1, Num of Neurons: 10	88.1±1.9%	0.8623

Conclusion

In this chapter, we have explored the use of integrating Fuzzy Logic with LSTM to classify physiological signals. It can be seen that the proposed methodology is able to achieve high classification accuracies for 4-classes. We will now look at improving the overall design further to incorporate other real-world characteristics of physiological signals.

Publication:

Fuzzy Inference System (FIS) - Long Short-Term Memory (LSTM) Network for Electromyography (EMG) signal analysis

Biomedical Physics & Engineering
Express 2022-10-27 | Journal article |
Author

DOI: [10.1088/2057-1976/ac9e04](https://doi.org/10.1088/2057-1976/ac9e04)

Part of ISSN: [2057-1976](https://doi.org/10.1088/2057-1976)

CONTRIBUTORS: Ravi Suppiah; Kim Noori; Anurag Sharma; Khalid Abidi

Chapter 6

BIO-inspired-Fuzzy- Inference-System (BIOFIS)

In this chapter, we introduce the novel Bio-Inspired-Fuzzy-inference-System. We will discuss the motivation behind the creation of this system, review its design and implementation details, and finally benchmark its performance across other Neuro-Fuzzy Inference Systems.

6.1 Motivation

The earlier chapters outlined several techniques employed to perform the various aspects of a complete system. These include signal processing, feature extraction, machine learning and classification. In terms of signal processing and feature extraction, several techniques have been explored: such as Discrete Wavelet Transform and Power Spectral Density. We have looked at fusing these signals together through Fuzzy Inference System with rules formed by observation and understanding of the underlying physical phenomena. The gap in these current techniques is that the features that are extracted from sensor data may not be consistent across different individuals. This is especially so when there are underlying physical and neuro-muscular issues, such as physical injury, stroke, lesions, etc.

The aim of BIOFIS is to:

- detect signals that are prominent based on each individual's underlying neuromuscular state
- extract and unify features across different data points
- select the most prominent features while discarding the rest to simplify the overall architecture
- auto-generate fuzzy-inference-rules based on these features
- employ short-term memory blocks to capture the temporal characteristics of the features

The overall design of the BIOFIS model is shown in Figure 6.1

6.2 Design & Implementation

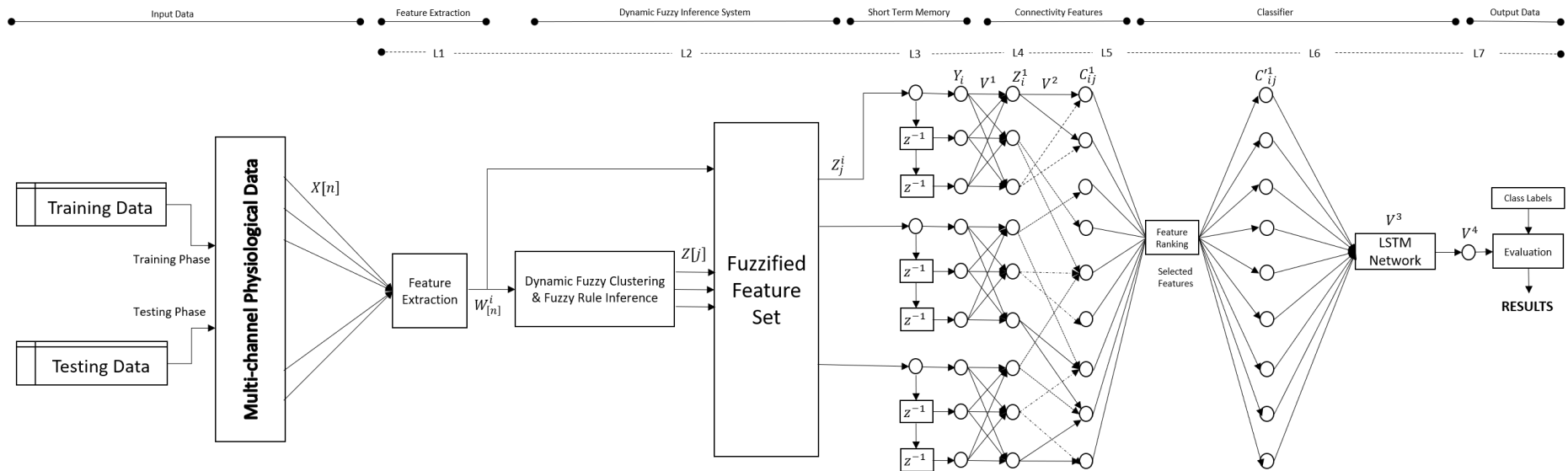


Figure 6.1: BIOFIS Model

The labels for the different signals of the system are as described in Table 6.1.

Table 6.1: Terms used in BIOFIS Model

$X[n]$	Multi-channel Physiological Data n represents the n -th channel.
$W_{[n]}^i$	Windowed feature set where n represents the channel and i represents the i^{th} window.
$Z[j]$	Fuzzy Clusters generated dynamically by the features.
Z_j^i	Fuzzified output window generated by $W_{[n]}^i$ input using $Z[j]$ clusters and inference rules
Y_i	Fuzzified output window. i represents the output of each fuzzy rule
V_{n_1, f_1}^1	weight between n^{th} node of 3rd layer and f^{th} hidden node of 4 th layer from ith channel
V_{f_i, f_j}^2	weights between f_i^{th}, f_j^{th} hidden node of 4 th layer and f_i^{th}, f_j^{th} hidden node of 5 th layer
C_{ij}^1	Weighted combination of inputs from all possible data streams
C'_{ij}	Selected features based on feature ranking
V^3	LSTM Network trained on selected features
V^4	Output Classifier Label
n_c	Number of channels of data
n_o	Number of samples per data vector

1st Layer – Signal Processing & Feature Extraction

An appropriate bandpass filter is applied to the input signals $X[n]$, to extract and focus on the frequency range of interest. For EEG signals, the range of 0 to 50Hz while for EMG signals, a range of 0 – 500Hz is used. After filtering, feature extraction techniques are employed. These include operations like Power Spectral Density (PSD), Wavelet Entropy (DWT), etc. The data that is obtained after the filtering and feature extraction, $\mathbf{W} \in \mathbb{R}^{n_p \times n_c}$, are used as inputs to the second layer.

2nd Layer – Dynamic Fuzzy Inference System

In this stage, we first perform Fuzzy C-means (FCM) clustering. FCM is a clustering method that allows each data point to belong to multiple clusters with varying degrees of membership. This allows us to naturally identify natural groupings of data from a large dataset to produce a representation that is more concise and relatable to the systems behaviour. To determine the optimal number of clusters, the Calinski-Harabasz clustering evaluation criterion is used. Well-defined clusters are those that generate a large between-cluster variance and a small within-cluster variance. The system will also automatically generate the fuzzy inference rules based on the clusters and the relationship between the input data and the output labels. In Fig 6.2, we can observe the membership functions automatically created using the clusters and the input/output data.

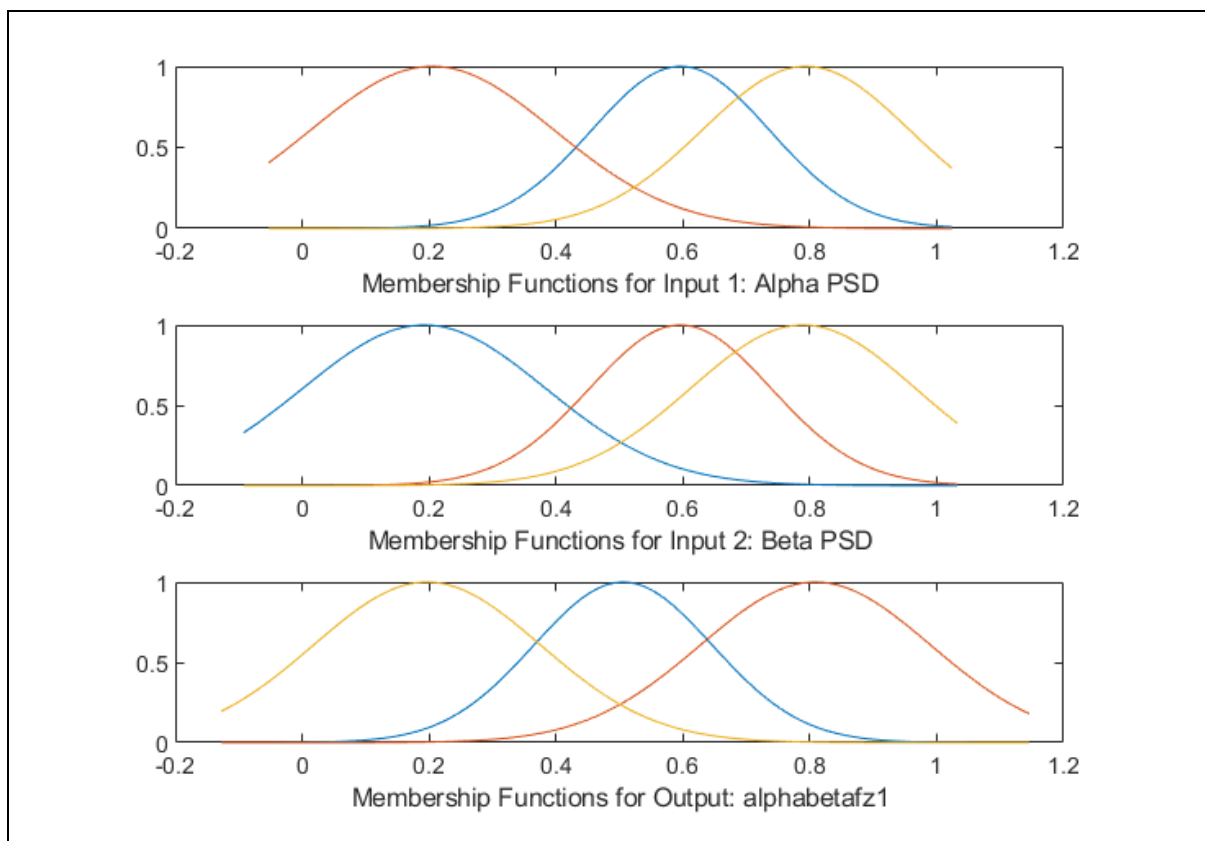


Fig 6.2: Generation of fuzzy inference rules based on the input/output data

3rd Layer – Short Term Memory

Physiological signals exhibit temporal relationships and the network must be able to capture these events to gain a better understanding of these signals. The proposed network, a serial-in, parallel-out shift register memory is used to keep the data for short window. The size of the window is configurable and can be set based on the characteristics of the signal being captured. The fuzzified data $Z[j]$ is passed to a rectangular window f_w to produce Z_j^i . The rectangular window function is computed using equations (6-1) and (6-2).

$$f_w(n) \triangleq \begin{cases} 1 & s_t \leq n \leq N \\ 0 & \text{otherwise} \end{cases} \quad \text{Eq. 6.1}$$

$$Z_i^j = f_w(n)Z[j]; \quad \forall n \in [s_t, N] \quad \text{Eq. 6.2}$$

Where Z_i^n denotes the n^{th} sample of the i^{th} channel of the windowed data and s_t is the start point of the window.

Z is represented as the input to the 3rd layer.

$$Z = [z_i, \dots, z_{n_c}] \quad \text{Eq. 6.3}$$

4th & 5th Layers – Connectivity Features

In this layer, we connect feature vectors formed from different windows across different input streams. This allows us to capture relationships across different sensors placed in different parts of the body.

However not all the extracted features (C_{ij}^1) are relevant and might have an adverse effect on the classifier. Furthermore, if there are a high number of features, then the next layer would be computationally heavy and inefficient. To reduce the number of selected features, the class separability criteria is used in this research.

The class separability factor for each feature is measured using the histograms of the feature vectors for both classes. The features are then sorted in descending order of the distances (larger distance = better separability = lower error). Following that, a significant number of features n_o are selected. The value of n_o is selected based on the largest gap in the sorted table.

The indices of the most significant features are then used to generate the output for the 5th layer using equation (6-4)

$$\mathbf{I} \in \{0,1\}^{n_o}, = \mathbf{I} = f_r(c', n_o), \quad \text{Eq. 6.4}$$

Where the f_r denotes the feature ranking function and \mathbf{I} denotes the indices of activated hidden nodes at the 4th layer. These selected features would then be the inputs for the next layer.

6th Layer – Classification

An LSTM network is a form of Recurrent Neural Network (RNN). RNN's are able to effectively learn short-term dependencies but they have an issue with long-term dependencies. LSTM's were designed to address the issue so that it can effectively learn both short- and long-term patterns in the data. The LSTM network will generate the output label V^4 .

7th Layer – Evaluation

The output label is then compared with the class labels to generate the classification results. Based on the results, the network will adjust the number of clusters dynamically and reevaluate the results. This is an iterative process until the most optimum number of clusters is obtained.

Learning & Testing

The learning process consist of the following stages.

Stage 1: Performing the necessary Signal Processing and Feature Extraction Techniques

Stage 2: Initiate the Fuzzy Network with 2 clusters and generate the inference rules.

Stage 3: Generate interconnections within signals formed from temporal differences within the same sensor and from different sensors.

Stage 4 & 5: Perform a ranking from the features and select the top N features as the once that demonstrate the most significant class separability.

Stage 6: Train the LSTM network based on the data generated from Stage 5

Stage 7: Evaluate the trained network.

6.3 Evaluation of BIOFIS based on Physiological Data

As mentioned in the earlier sections, the physiological data is obtained from existing dataset. The signal processing and feature extraction techniques are as what has been described before. The performance of the BIOSFIS network was compared against other networks and techniques. The results for EEG data are shown in Table 6.2 and those for EMG data are shown in Table 6.3.

Table 6.2: Summary of Feature Selection and Classification Techniques for EEG Data

Classifier – Feature Selection	Classification Accuracy		
	<i>Accuracy (%)</i>	<i>Precision (%)</i>	<i>F1-score</i>
BIOFIS – ERD/ERS	98.16±1.15	97.76±1.62	0.9782
BIOFIS – PSD	95.12±1.11	94.98±1.36	0.9691
BIOFIS – Moving Average	89.47±0.37	88.41±1.43	0.8981
BIOFIS – Wavelet Coefficients	85.34±0.75	83.28±1.28	0.8412
kNN – ERD/ERS	89.18±0.69	89.13±0.94	0.8807
kNN – PSD	90.51±1.39	88.56±1.02	0.8942
kNN – Moving Average	85.28±0.73	83.12±1.19	0.8476
kNN – Wavelet Coefficients	80.16±1.15	78.34±1.34	0.7865
SVM – ERD/ERS	91.23±1.29	90.25±1.11	0.8932
SVM – PSD	89.51±1.86	89.14±0.37	0.8734
SVM – Moving Average	86.28±0.27	82.34±0.74	0.8595
SVM – Wavelet Coefficients	80.25±0.41	80.06±0.59	0.7805
LSTM – ERD/ERS	92.91±1.43	90.43±1.22	0.9105
LSTM – PSD	90.45±1.07	89.91±0.56	0.8804
LSTM – Moving Average	87.45±0.97	87.23±0.11	0.8602

LSTM – Wavelet Coefficients	85.22±0.45	83.15±0.24	0.8498
ANFIS – ERD/ERS	90.48±0.69	90.12±0.73	0.8909
ANFIS – PSD	89.93±0.45	88.78±1.68	0.8843
ANFIS – Moving Average	88.25±1.17	87.25±1.25	0.8751
ANFIS – Wavelet Coefficients	87.78±1.22	86.23±1.13	0.8673

Table 6.3: Summary of Feature Selection and Classification Techniques for **EMG** Data

Classifier – Feature Selection	Classification Accuracy		
	<i>Accuracy (%)</i>	<i>Precision (%)</i>	<i>F1-score</i>
BIOFIS – RMS	97.18±1.16	97.09±1.64	0.9742
BIOFIS – PSD	93.22±1.30	94.03±0.82	0.9471
BIOFIS – Moving Average	89.47±0.98	88.41±0.98	0.8893
BIOFIS – Wavelet Coefficients	84.49±0.74	82.45±1.35	0.8314
kNN – RMS	90.18±0.98	89.13±0.42	0.8945
kNN – PSD	90.51±1.17	87.67±1.29	0.8975
kNN – Moving Average	85.28±1.24	83.12±1.17	0.8431
kNN – Wavelet Coefficients	78.56±1.57	77.56±1.78	0.7793
SVM – RMS	90.32±0.51	89.35±1.13	0.8909
SVM – PSD	87.72±0.73	87.36±0.80	0.8702
SVM – Moving Average	85.42±1.82	81.52±0.41	0.8352
SVM – Wavelet Coefficients	79.45±0.15	79.26±0.14	0.7906
LSTM – RMS	91.92±0.60	90.43±1.96	0.9103
LSTM – PSD	90.45±1.74	89.01±1.73	0.8904
LSTM – Moving Average	87.45±1.29	86.36±1.58	0.8675
LSTM – Wavelet Coefficients	85.22±0.96	83.15±0.55	0.8464
ANFIS – RMS	88.67±0.68	88.32±0.53	0.8892
ANFIS – PSD	86.03±0.22	86.89±0.91	0.8678
ANFIS – Moving Average	87.37±1.57	87.25±1.17	0.8719
ANFIS – Wavelet Coefficients	89.03±1.66	88.51±1.29	0.8892

For each classifier the feature that generated the highest accuracy is extracted for easier comparison. In the EEG domain the extracted combinations are: BIOFIS - ERD/ERS (98.16±1.16), kNN - PSD (90.51±1.39), SVM - ERD/ERS (91.23±1.29), LSTM – ERD/ERS (92.91±1.43), ANFIS – ERD/ERS (90.48±0.69). These results are shown in Figure 6.3.

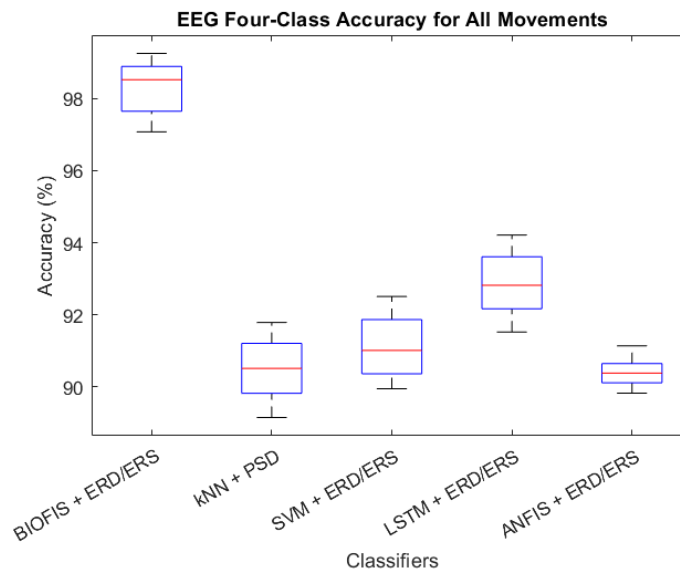


Fig 6.3: EEG Four-Class Accuracy for All Movements

In the EMG domain, the extracted combinations are: BIOFIS – RMS(97.18±1.16), kNN-PSD (90.51±1.17), SVM – RMS (90.32±0.51), LSTM – RMS (91.92±0.60), ANFIS – Wavelet Coefficients (89.03±1.66). These results are shown in Figure 6.4.

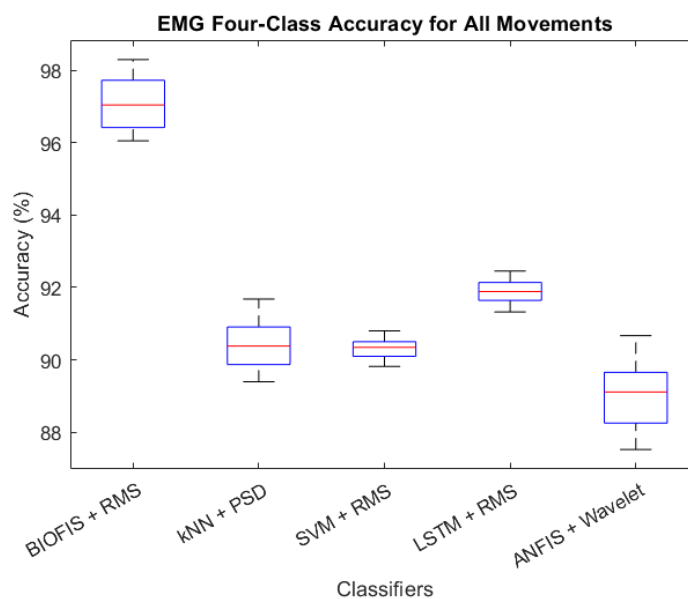


Fig 6.4: EMG Four-Class Accuracy for All Movements

6.4 Applications of BIOFIS

One of the key requirements for a rehabilitative system is to be able to adapt itself to individuals who may have physiological signals impaired due to many reasons such as stroke, lesions, injury, etc. The testing and results of BIOFIS demonstrate that it is able to outperform other models built using a combination of feature-sets and machine learning techniques.

A feature ranking is first performed using the Minimum Redundancy Maximum Relevance (MRMR) algorithm. This algorithm aims to find an optimal feature set that is both mutually and maximally dissimilar, while being able to effectively represent the response variable effectively. The feature ranking is as shown in Figure 6.5.

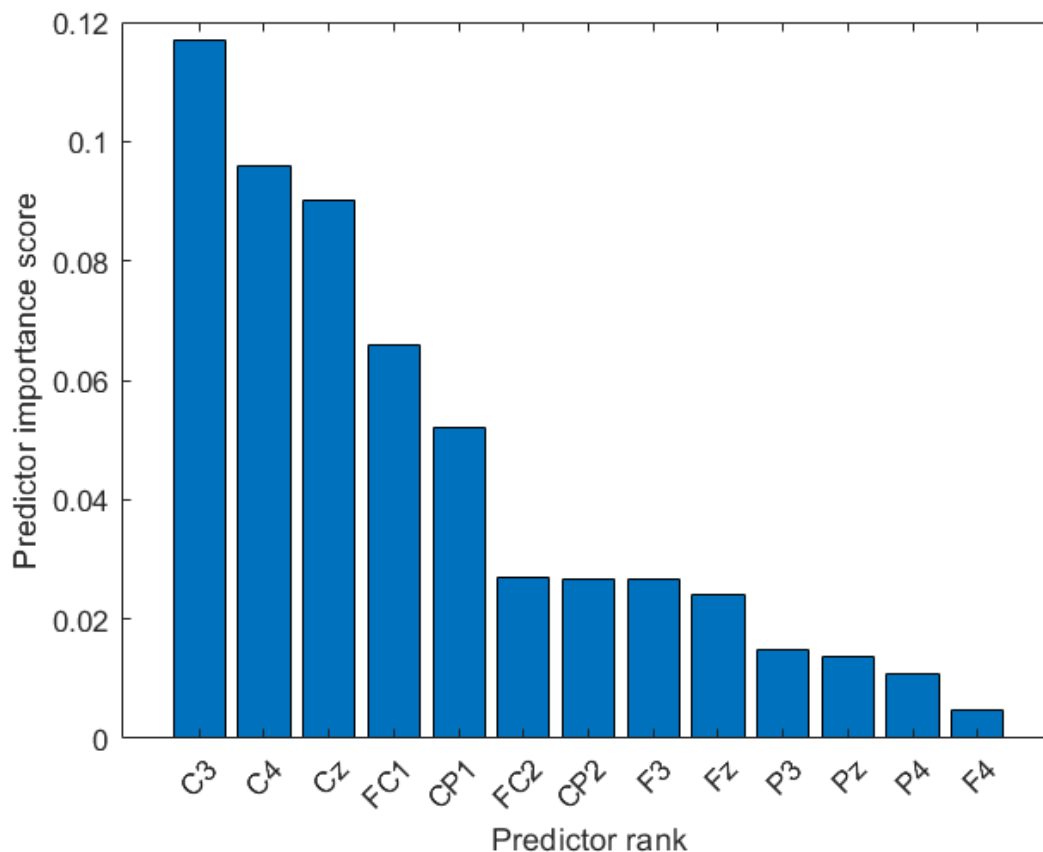


Figure 6.5: Feature Ranking of topmost EEG features

It can be seen that the first five predictors, Channels C3, C4, Cz, FC1 and CP1 have the greatest predictor importance scores for the EEG features.

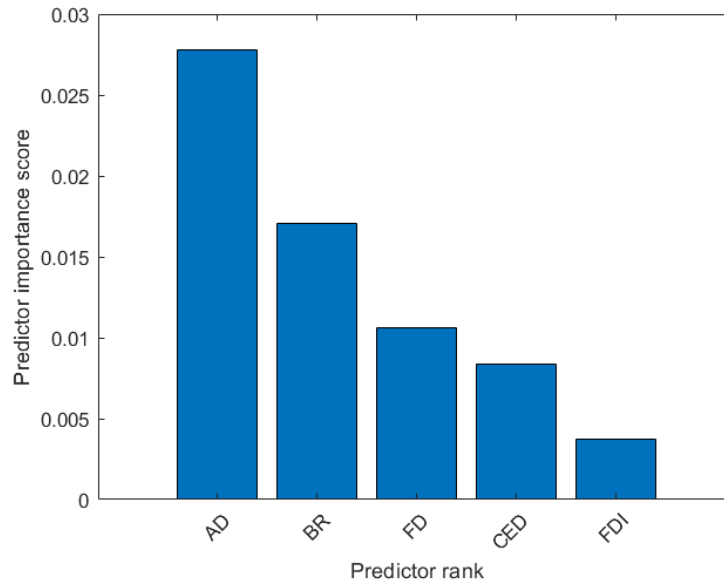


Figure 6.6: Feature Ranking of EMG features for the Forward/Reverse Movement

The feature ranking for the EMG data is shown in Figure 6.6. It can be seen that the AD channel has a significantly higher importance than the rest of the channels.

6.5 Reduced Physiological Signals

In order to measure the performance of the system in an environment where the signals are diminished we selected the top-ranking feature and removed it from the system in both the training and testing phase. For the EEG domain, the C3 channel was removed and for the EMG domain, the AD channel was removed. Figures 6.7 shows the classification accuracy for these cases.

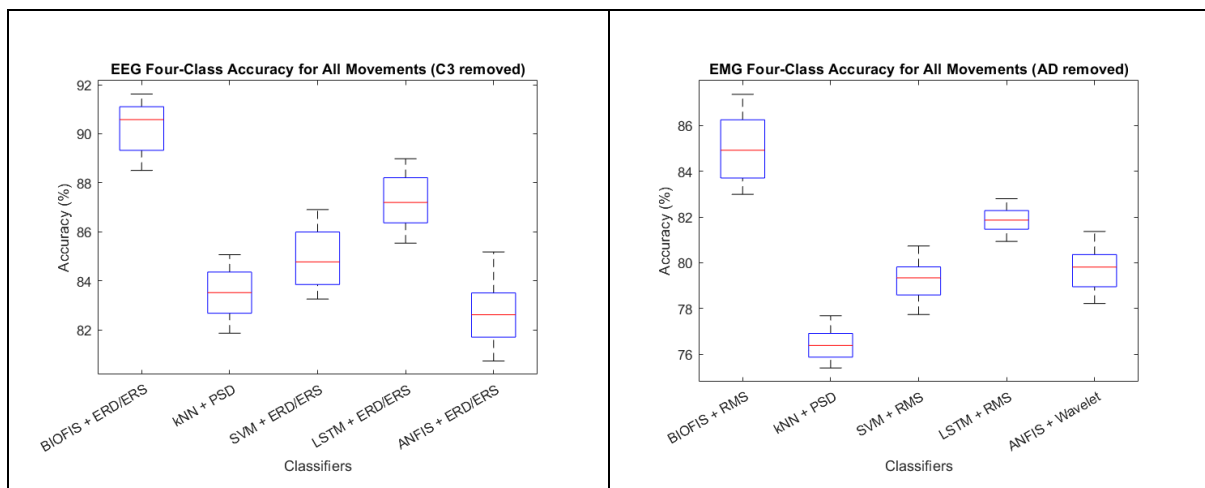


Figure 6.7: Classification accuracy with Key Feature Reduced

In both cases, it can be seen that the BIOFIS model outperforms the other classifiers in the reduced feature set experiment. In the EEG domain the classification accuracy dropped to $90.06 \pm 1.66\%$, a drop of about 8% from the full feature set. In the EMG domain, the classification accuracy dropped by around 12% to $85.18 \pm 2.26\%$. In the EEG feature set, there were other channels beside C3 with a high predictor importance score, and this allowed the system to still retain a high accuracy score with the reduced feature set. In the EMG feature set, the AD channel had a significantly higher importance score compared to the channels. As such, the overall accuracy dropped further as compared to the EEG domain. These results demonstrate that the BIOFIS system is able to naturally adapt itself to the nature of the physiological signals that it is exposed to without being overly reliant on any particular group of signals. In a rehabilitative setting, such a system will be able to learn the limitations of the signals generated by individuals and train itself to maximize its classification accuracy using the available signals.

6.6 Alternate Dataset

NinaPro-DB7

Myoelectric and Inertia (IM) data was collected from 12 sensors placed on the arm. Eight sensors were equally spaced around the forearm, two were placed to target the extrinsic hand muscles, and the other two were placed on the biceps and triceps muscles. Data were collected from 20 subjects who performed a series of 40 motions, including individuated-finger, hand, wrist, grasping and functional movements. Each movement was repeated six times and trials were interleaved with sufficient resting periods. Using sEMG signals alone, a classification accuracy of $60.0 \pm 11.5\%$ was achieved across all subjects and classes in [215].

NinaPro-DB1

sEMG and kinematic signals of the hand and wrist were obtained from 27 healthy subjects performing a total of 52 hand movements. These include basic finger movements, isometric and isotonic hand configurations and grasping and functional movements. Eight active electrodes are uniformly placed around the forearm. Two additional sensors are placed on

the large flexor and extensor muscles of the forearm. An average classification accuracy of $75.7 \pm 4.5\%$ was achieved across all subjects and classes [216].

Assessment of BIOFIS with Alternate Datasets

The BIOFIS model was tested using these datasets with the various combinations of the model with different combinations of feature extraction methods. The RMS and PSD features produced this highest classification accuracies as shown below:

NinaPro-DB7: RMS -> $78.9 \pm 1.6\%$, PSD -> $75.1 \pm 2.7\%$

NinaPro-DB1: RMS -> $87.3 \pm 2.3\%$, PSD -> $83.7 \pm 1.9\%$

The results, as summarized in Table 6.4, demonstrate that the proposed model is able to outperform the state-of-the-art.

Table 6.4: Highest Classification Accuracies for Alternate Datasets

	RMS	PSD
NinaPro-DB7	$78.9 \pm 1.6\%$	$75.1 \pm 2.7\%$
NinaPro-DB1	$87.3 \pm 2.3\%$	$83.7 \pm 1.9\%$

The results demonstrate that the proposed model is able to outperform the state-of-the-art.

Ablation Study

It can be seen from the results that the proposed BIOFIS model is able to consistently outperform existing methods with high levels of classification accuracy. To further understand the significance of the model, we perform an ablation study to understand its significant contributions.

The BIOFIS model has 3 main sections that be used in this study. They are

- Fuzzification of Input Features
- Short-Term Memory and Connectivity Features
- Feature Ranking

We proceed to remove each of these sections individually to understand its impact on the overall performance of the system.

In the removal of the fuzzification component, we directly feed the feature set into the short-term memory blocks of the system which are then processed as before. In the case of the short-term memory and connectivity features, we directly feed the fuzzified features into the feature ranking block. When we remove the feature ranking block, the connectivity features are sent directly to the LSTM block. Tables 6.5 and 6.6 show the results of this study for EEG and EMG features respectively. We use the features that generated the highest classification accuracy for the complete BIOFIS model.

Table 6.5: BIOFIS Ablation Study for EEG Features

<i>Classifier – Feature Selection</i>	<i>Classification Accuracy</i>	
	<i>Accuracy (%)</i>	<i>Precision (%)</i>
BIOFIS – ERD/ERS	98.16±1.15	97.76±1.62
BIOFIS – ERD/ERS with fuzzification removed	72.32±3.15	71.85±2.47
BIOFIS – ERD/ERS with short-term memory and connectivity features removed	71.56±1.21	72.82±1.42
BIOFIS – ERD/ERS with feature ranking removed	83.63±2.49	82.18±0.56

Table 6.6: BIOFIS Ablation Study for EMG Features

<i>Classifier – Feature Selection</i>	<i>Classification Accuracy</i>	
	<i>Accuracy (%)</i>	<i>Precision (%)</i>
BIOFIS – RMS	97.18±1.16	97.09±1.64
BIOFIS – ERD/ERS with fuzzification removed	74.86±2.05	73.19±0.92
BIOFIS – ERD/ERS with short-term memory and connectivity features removed	73.94±1.09	73.94±3.13
BIOFIS – ERD/ERS with feature ranking removed	84.81±1.73	83.26±1.92

The study is repeated with reduced physiological signals, as described in Section 6.5. The results are as shown in Tables 6.7 and 6.8.

Table 6.7: BIOFIS Ablation Study for EEG Features with C3 removed

<i>Classifier – Feature Selection</i>	<i>Classification Accuracy</i>	
	<i>Accuracy (%)</i>	<i>Precision (%)</i>
BIOFIS – ERD/ERS	90.06±1.66	90.56±1.62
BIOFIS – ERD/ERS with fuzzification removed	65.47±2.06	62.98±1.99
BIOFIS – ERD/ERS with short-term memory and connectivity features removed	63.78±2.76	64.15±1.46
BIOFIS – ERD/ERS with feature ranking removed	56.78±1.48	53.97±0.93

Table 6.8: BIOFIS Ablation Study for EMG Features with AD removed

<i>Classifier – Feature Selection</i>	<i>Classification Accuracy</i>	
	<i>Accuracy (%)</i>	<i>Precision (%)</i>
BIOFIS – RMS	85.18±2.26	86.01±0.74
BIOFIS – ERD/ERS with fuzzification removed	64.35±1.36	61.92±0.38
BIOFIS – ERD/ERS with short-term memory and connectivity features removed	62.83±1.04	62.58±2.39
BIOFIS – ERD/ERS with feature ranking removed	55.24±0.93	54.12±1.82

6.7 Discussion

The BIOFIS model has been shown to be able to generate high classification accuracies across different datasets. It outperformed existing techniques by a significant margin. This performance can be attributed to the three main blocks of the model. Firstly, it is the dynamic fuzzy clustering model that generates a fuzzy output from the features. The rule-base used in the process closely resembles our understanding of the physiological signals and their behavior. With fuzzified signals, we are able to better represent the nuances of the various

channels in a more natural way. Next, the short-term memory block and the connectivity features mimic the natural flow of the physiological signals in our body. In our body, anytime a neuron fires, it activates other neurons around it within a time-window. The firing strength of each neuron is dependent on many other neurons around it. This natural phenomenon is replicated by this block as it establishes a relationship between various nodes. Finally, the feature ranking block allows the model to rank features based on their significance. This ensures that only those features that are important affect the final outcome. These unique features of the BIOFIS model have enabled it to consistently generate high accuracy results.

The ablation study reveals some important characteristics about the model. With the full set of features, the feature ranking feature has a lesser effect on the classification accuracy. The accuracy dropped to around 83%-84% with the feature ranking removed, compared to the range of 71%-72% with the fuzzy clustering and short-term memory sections removed. However, in the case of the reduced feature set, we can see that the feature ranking feature had the most impact on classification accuracy, dropping to the range of 54%-56% when it was removed, compared to 62%-64% when the fuzzy clustering and short-term memory sections removed. This shows that while all three blocks are critical, the feature-ranking block plays a significant role, especially in the real-world scenario where the user may have a reduced set of features due to underlying physiological conditions.

6.8 Conclusion

In Chapters 3 to 6, we have developed advanced models to analyze and efficiently classify physiological signals to decode the user's intention. The high classification accuracies obtained from real-world scenarios demonstrate the effectiveness of the proposed methods. These results have been obtained through the use of high-powered computing systems with extensive resources. In a practical rehabilitative device, the solution would need to be compact and built into a standalone system that is portable. This requires the use of an embedded computing platform that would be able to perform advanced signal processing and machine learning algorithms. We will study this approach in the next chapter.

Chapter 7

Edge Computing

7.1 Overview

In this final chapter, we will look at applying signal processing and machine learning techniques onto an embedded platform. This step allows to realize the potential of porting complex algorithms onto a device that can act on its own without the need for cloud connectivity. It allows us to create a stand-alone rehabilitative device that can really benefit from the extensive research in this area.

Artificial Intelligence is an umbrella term for all the studies that aim to replicate human reasoning with computer systems. You use aspects of this technology every day in a wide range of applications, such as facial recognition and voice recognition, autonomous driving, stock market trading, disease symptom detection, predictive maintenance, handwriting recognition, content distribution on social media, detection of fraudulent credit card transactions, machine translation and shopping recommendations. Machine Learning is a subset of Artificial Intelligence that refers to a computerized system capable of using relevant data to improve at a specific task. Machine Learning involves two steps, the training step and the inference step. During the training step, the machine uses past or known data to train a model and calculate the "learning" parameters of the model. During the prediction or inference step, the machine uses the trained model with the trained parameters to make predictions from new data that were unavailable when the machine was trained. This new data could be arriving in real time, for example as new images from a camera or as an audio stream from a microphone. In supervised learning, labeled datasets — such as images of fruit labeled with the type of fruit — are used to train the models. The inference step uses the previously trained model and predicts the output, ideally in real time. The model can also

include the feature extraction that is processed on the input data. Usually, the training step must be performed on a desktop computer as it is the most computationally intensive task. On the other hand, the inference step can be performed on a smaller, low-powered processor such as the Arm Cortex-M family processors that will be used in this course. In this case, the Machine Learning is "at the edge" as opposed to "in the cloud".

7.2 Design & Implementation

The system will comprise of both Hardware (HW) and Software (SW) implementation. We will be using an ARM-based microcontroller that has the required processing power and memory to perform the required machine learning algorithm.

Hardware Setup

The STM32L496ZGTx is used as the microcontroller chosen for the design. It belongs to a class of low-power microcontrollers based upon the high-performance 32-bit ARM Cortex-M4 processors. Its built-in floating-point unit (FPU) makes signal processing computations more effective. Its internal memory of 320 Kbyte of SRAM and 1 Mbyte of Flash provide adequate memory for many Edge-Computing applications to host its own neural network. Its wide range of peripheral devices like Analog-to-Digital Converters (ADCs), General Purpose Timers, Pulse-Width Modulation (PWM) blocks and many others provides adequate subsystems for a complete single-chip solution.

The STM32 microcontroller is the main device that will perform all the necessary computations. These include the signal processing, machine learning, and actuator control. To compare the performance of the embedded solution to the offline solution implemented in Matlab, the same dataset is used. Instead of performing live data capture using the EMG sensor, the samples associated with each trial are sent from the PC to the STM32 through a serial interface. The results of the classifier are sent back through the serial interface.

Software Design

The software design has several steps as we need to train the neural network weights and interconnections first before deploying them onto the embedded platform.

7.3 Training and Finalizing the Network

Step 1: Tensorflow

We need to install Python and several associated support packages on the PC. These include pandas, jupyter, scikit, tensorflow, and matplotlib. These packages are needed to ensure that we can train and develop the required machine learning models using Python.

Step 2: Load Dataset

With the necessary packages installed, we proceed to load the dataset into the environment.

```
letter = input('Please insert letter to collect data: ')
stride = 30
f = os.path.join(samples_dir, 'letter_{}_stride_{}.csv'.format(letter, stride))
if os.path.exists(f):
    print('File exists and data will be appended...')
    xyz_df = pd.read_csv(f)
else:
    print('New sample, starting blank...')
    xyz_df = pd.DataFrame(columns=['X', 'Y', 'Z'])

while input('1 - acquire sample, 2 - exit: ') == '1':
    line = ser.readline()
    lineList = convert_to_list(str(line))
    new_df = convert_list_to_df(lineList)
    print('New data acquired:\n', new_df.describe())
    xyz_df = pd.concat([xyz_df, new_df], ignore_index=True)
    print('Total Data count:', int(xyz_df.shape[0]/stride))

print('Saving data to:', f)
print('Total data of sample {}:\n'.format(letter), xyz_df.describe())
xyz_df.to_csv(f, index=False)
```

Step 3: Create and train a Multi-Layer Perceptron Model

```
x_train, y_train = sklearn.utils.shuffle(np.array(data), np.array(labels))
y_train = tf.keras.utils.to_categorical(y_train, len(np.unique(y_train)))

model = tf.keras.Sequential(
    [
        tf.keras.Input(shape=(3, stride)),
        tf.keras.layers.Flatten(),
        tf.keras.layers.Dense(30, activation="relu"),
        tf.keras.layers.Dropout(0.5),
        tf.keras.layers.Dense(20, activation="relu"),
        tf.keras.layers.Dropout(0.5),
        tf.keras.layers.Dense(len(np.unique(y_train)), activation="softmax")
    ]
)

model.summary()
```

Step 4: Train the Model

```
model.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=['accuracy'])
history = model.fit(x_train, y_train, batch_size=1, epochs=200,
validation_split=0.4)

model.save('raw_model.h5')
```

Step 5: Test the Model

```
input('Press Enter once MCU is ready')
line = ser.readline()
lineList = convert_to_list(str(line))
new_df = convert_list_to_df(lineList)
print('New data acquired:\n', new_df.describe())

x = new_df['X'].to_numpy()
y = new_df['Y'].to_numpy()
z = new_df['Z'].to_numpy()

inf_data = np.array([x, y, z])
plot_single_sample(data_sample=inf_data.reshape((3, stride)))

# For inference we have to explicitly tell that the data has a batch size of 1
inf_data = inf_data.reshape((1, 3, stride))

pred = model.predict(inf_data)
print('Model Prediction: ', np.argmax(pred))
```

Step 6: Save the Model

Once the model has been trained and tested, we can save the model

7.4 MCU Deployment

Validation

Using STM32 Cube AI and STM32 Cube IDE, we will be able to deploy the model onto the STM32 embedded board. Figure 7.1 shows the STM32 microcontroller on the board.

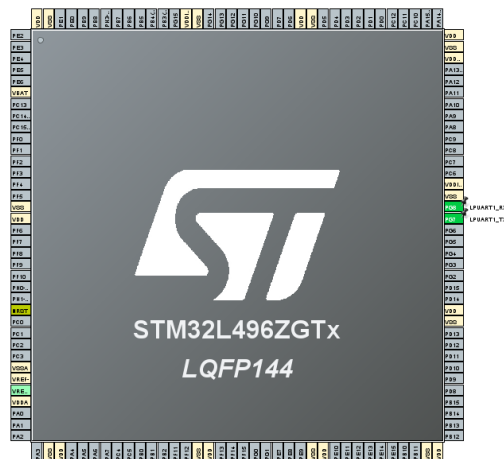


Fig 7.1: Pin-Out Configuration on the STM Cube MX

In the STM Cube MX IDE, we will be able to perform the necessary pinout configuration as well as install the required Cube AI Software Pack. With that, we can upload the model that we created in Jupyter Notebook and validate it.

The validation process is shown in Figure 7.2. Once validated, we can proceed to implement an application with the model.

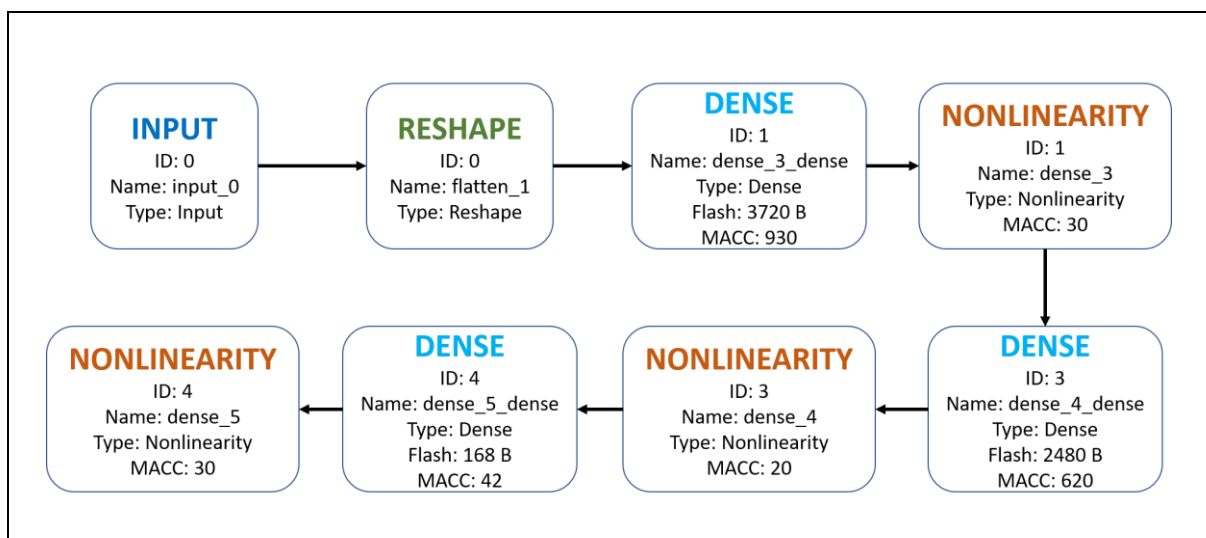


Fig 7.2: Validation Analysis on the MCU Target

Application Development

Once the model has been validated, we can implement an application with the model. Regenerating the code from STM Cube AI, an application file 'app_x-cube-ai.c' will be generated. Within this file, we will be able to perform the necessary data acquisition, signal processing and feature extraction.

7.5 Testing the Network

In order to gauge the performance of the network on the embedded board, raw data for each participant is extracted from the dataset and sent to the board through the serial interface. The application code on the board will perform the various signal processing and feature extraction techniques before passing on the data to the trained network. The classification result from the network is displayed on the board and PC.

Phase 1

In the first instance, the algorithm as described in [217] is developed and deployed on the MCU. In it the raw signal is first pre-processed with a bandpass filter of 10Hz – 500Hz followed by a Root-Mean-Square (RMS) extraction to smoothen the signal. In the next phase, the Discrete Wavelet Transform is performed on the signal to generate the detailed and approximate filter coefficients. Using these, a final decomposition vector is generated. The Entropy value for vector is then computed as the final data point for the classifier. Figure 7.3 shows the confusion matrix plot for the AD channel for two different pairs of hand movements.

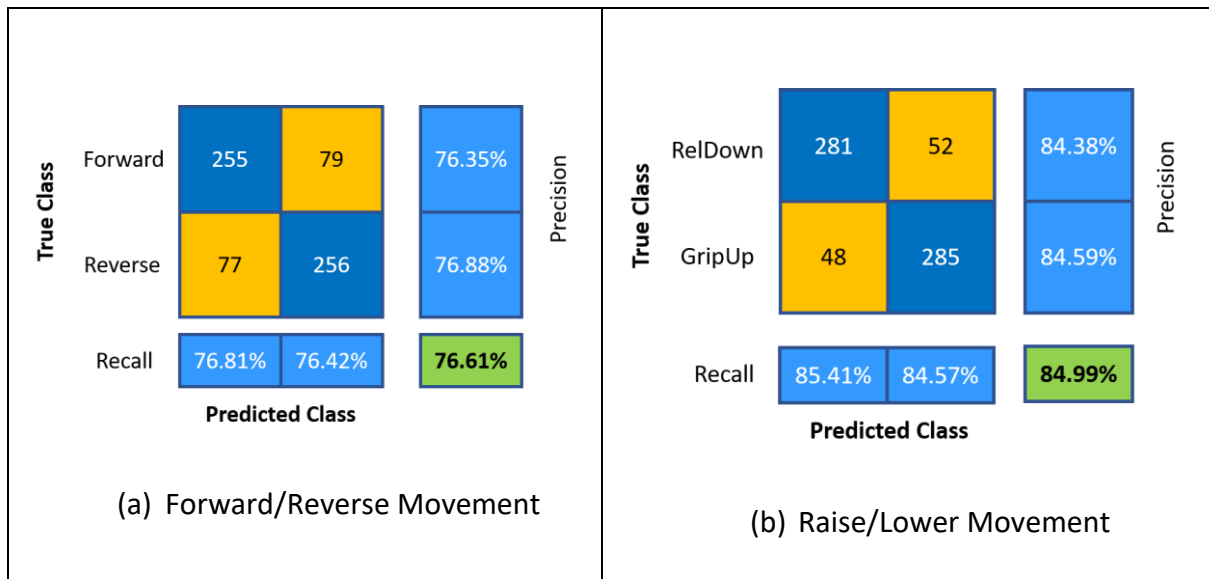


Fig 7.3: Confusion Matrix for two different pairs of movements

Figure 7.4 shows the classification accuracies for the various channels based on these two pairs of movements. A comparison is made with the results obtained from the paper.

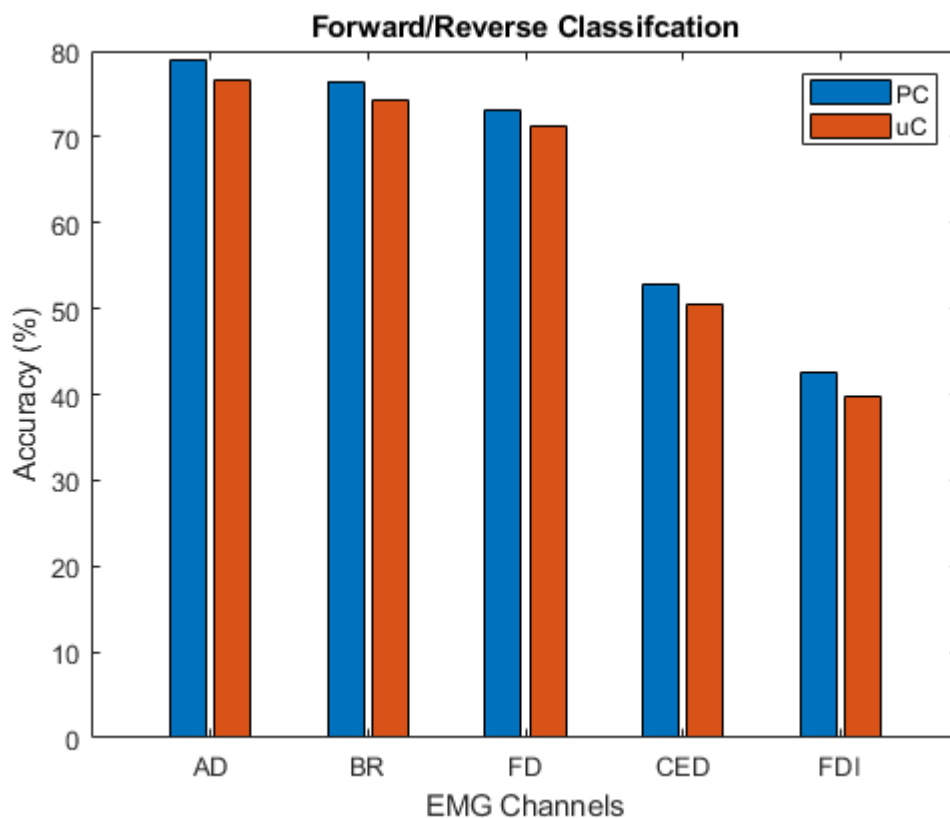


Figure 7.4: Comparison of Classification Accuracies between the PC-based system (PC) and the Embedded Controller (uC).

When porting a Neural Network architecture from a PC-based solution like Matlab or Python over to an embedded platform, the accuracy will naturally be lower. There are two main reasons for this. Firstly, the PC-based platform runs on a 64-bit processor and the resolution of the weights used in the network are higher. Secondly, the STM32 Cube AI needs to generate a network that can be compressed within the resource constraints of the controller. As such, it does auto-pruning of weights that have a low value so as to strike a trade-off between resources and accuracy.

In the paper, the authors observed that different channels had a varied effect on the overall accuracy of the system. The AD, CED and FDI channels were more prominent in the Raise and Lower actions while the AD, BR and FD channels played a greater role in the Forward and Reverse actions. Based upon this observation, the authors proposed a novel Unified Classification model that made use of TWO separate Neural Networks trained using these different combinations of channels. As a result, the overall accuracy of the system increased.

FIVE-WAY Real-Time Classification

The solution of two distinct neural networks is feasible in a PC-based environment. However, it is highly impractical in an embedded solution. Resource-Constraint Microcontrollers have the ability to at most host a single network. As such, the solution proposed by the authors cannot be directly applied in the context of embedded solutions. There have to be improvements in the areas of signal processing and feature extraction to solve this challenge.

7.6 Improved Methodology

Proposed Algorithm using Cepstral Analysis

Cepstral[218] analysis is tool that allows us to analyze the periodic segments of a signal within its frequency range. It is a nonlinear operation that has wide range of applications in image, speech and other types of signals. Though there are many variants, the most common ones are the real, complex and power cepstrum.

Real Cepstrum is defined as the inverse Fourier transform of the real logarithm of the magnitude of the Fourier transform of a sequence.

$$C_r = \mathcal{F}^{-1}\{\log(|\mathcal{F}\{f(t)\}|)\} \quad \text{Eq. 7.1}$$

The output generated by performing the cepstrum analysis provides us with information about the rate of change that is observed in the various spectrum bands.

Discrete Wavelet Transform

Prior to performing the Cepstrum Analysis, the Discrete Wavelet Transform is first performed. In DWT, the wavelets are sampled discretely. They offer the same advantage over the Fourier Transform by offering time-frequency localization information. It has been used extensively in many applications, and two popular domains are data compression and filtering. A wide range of wavelets have been proposed and used, and the most common ones are the Haar wavelet and the Daubechies wavelet.

We can compute the DWT of a signal x by passing it through a filter series. At the first stage, the signal is convoluted with an impulse response g and passed through a low pass filter as shown in Equation (7.1), where n , denotes the length of the filter, y , refers to the output samples and k represents the amount of shifting applied to the signal as part of the convolution process.

$$y[n] = (x * g)[n] = \sum_{k=-\infty}^{\infty} x[k]g[n - k] \quad \text{Eq. 7.2}$$

A high-pass filter h is also used to decompose the signal simultaneously. We then obtain two outputs simultaneously. They are the output form of the high-pass filter, also known as the detail coefficients. We also have the output from the low-pass filter, the approximation coefficients. We are then able to perform subsampling by 2 by passing the signal through new filters g and h , as whosn in Equations (7.2) and (7.3), with the cut-off frequency being halved as compared to the previous one.

$$y_{low}[n] = \sum_{k=-\infty}^{\infty} x[k]g[2n - k] \quad \text{Eq. 7.3}$$

$$y_{high}[n] = \sum_{k=-\infty}^{\infty} x[k]h[2n - k] \quad \text{Eq. 7.4}$$

This decomposition technique is continuously repeated to allow us to have a higher frequency resolution, and this is represented using a binary tree that shows the time-frequency localisation. This binary tree is also referred to as the filter bank.

For any given signal s that has a length of N , the Discrete Wavelet Transform (DWT) will be made up of $\log_2 N$ steps. We start with the original signal s , and the first step generates two sets of coefficients, the Detail Coefficients cD_i and the Approximation Coefficients cA_i . This is obtained by performing a convolution with the high-pass filter HiD , and the low-pass filter LoD . The results of the filtering operation are then down-sampled to obtain cD_i and cA_i . This is shown in the figure below.

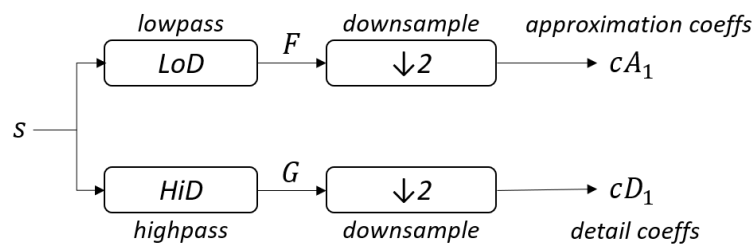


Figure 7.5: DWT operation

By reconstructing the signal from the Detailed Coefficients, we focus on the high-frequency changes in the signal that represent the patterns related to the user's movements that we intend to capture and classify.

Root-Mean Square

As the results of the Cepstrum Analysis contain both positive and negative values, the Root-Mean-Square (RMS) is performed to convert all signals to positive values.

The overall flow of the proposed algorithm is as shown in Figure 7.6.

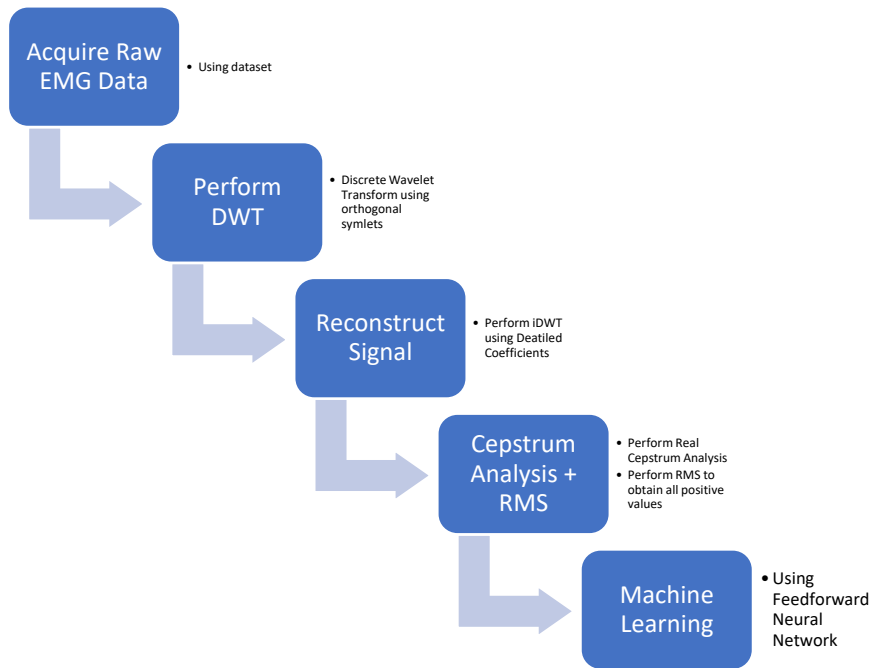


Fig 7.6: Flow of Proposed Algorithm

Real-Time Software Architecture

The embedded software is designed using the FreeRTOS [219] framework. FreeRTOS is an open-source Real-Time Operating System designed for use in the embedded domain. It includes a kernel and several libraries that can be used across a wide range of industrial applications. Due to its open-source nature, there are continuous improvements and upgrades to the libraries and APIs through contributions from the community.

The overall architecture is as shown in Figure 7.7. There is a total of EIGHT tasks and 1 Timer Interrupt. The objectives of each of these are as follows:

Timer Interrupt:

The Timer block is configured to generate a periodic 250ms interrupt. In the Interrupt Service Routine (ISR), the `osSemWin()` is released. `osSemWin()` is a Binary Semaphore initialized with a starting value of 0 in `main()`.

tWindow:

The `tWindow()` task starts by attempting to acquire the `osSemWin()` semaphore. Since the initial value is 0, it goes to a BLOCKED state during the first run. When the Timer ISR is executed, the Semaphore is released and this UNBLOCKS the `tWindow()` task. Once

unblocked, the task proceeds to capture a 500ms window of the raw data. This data is then posted to the tDWT_MessageQueue.

tDWT:

This task is in a BLOCKED state until data arrives on its internal message queue, tDWT_MessageQueue. Once a new data packet arrives, it will perform the DWT operation on the packet and then reconstruct the signal using the Decomposition Vectors. This output is then posted to the tCepstrum_MessageQueue.

tCepstrum:

This task is in a BLOCKED state until data arrives on its internal message queue, tCepstrum_MessageQueue. Once a new data packet arrives, it performs the Cepstrum Analysis on the DataPacket. Following that, the RMS operation is performed. The output is then posted to the tANN_MessageQueue.

tANN:

This task is in a BLOCKED state until data arrives on its internal message queue, tANN_MessageQueue. Once a new data packet arrives, it performs the classification using the model trained and deployed earlier using the STM32 Cube AI. Based on the classification result, the appropriate Semaphore is released to perform the appropriate motor movement.

tForward, tReverse, tGrip and tRelease:

These tasks work in a similar way, they are BLOCKED until the appropriate Semaphore is released from tANN. osSemForward() unblocks tForward, osSemReverse() unblock, tReverse, osSemGrip() unblocks tGrip, and osSemRelease() unblocks tRelease. When any of these tasks is unblocked, the mototMutex() is first acquired. This mutex allows exclusive access to the motors to ensure that once a motor movement has been started, it will run to completion before starting the next move.

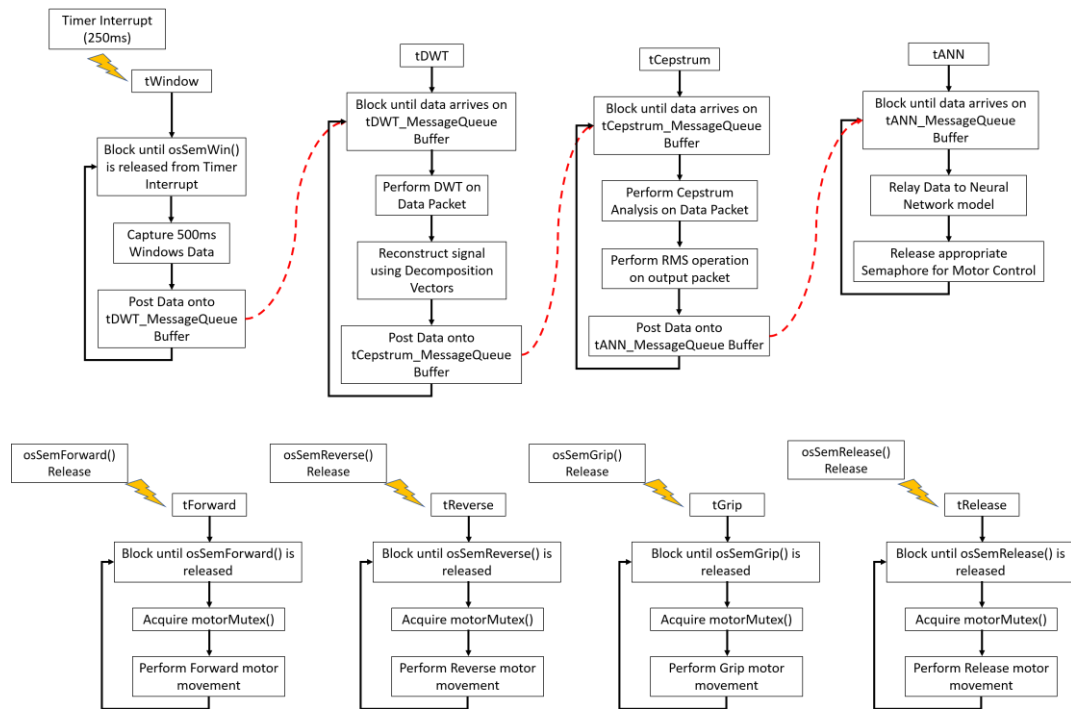


Fig 7.7: RTOS Architecture

Testing and Results

Figure 7.8 shows the scatter plot from the features generated by the proposed algorithm. It can be seen that using just two channel, AD and BR, there is a good distinction between the various actions performed by the users with minimal overlap. This allows us to achieve an average classification accuracy of 97.51% for all four movements.

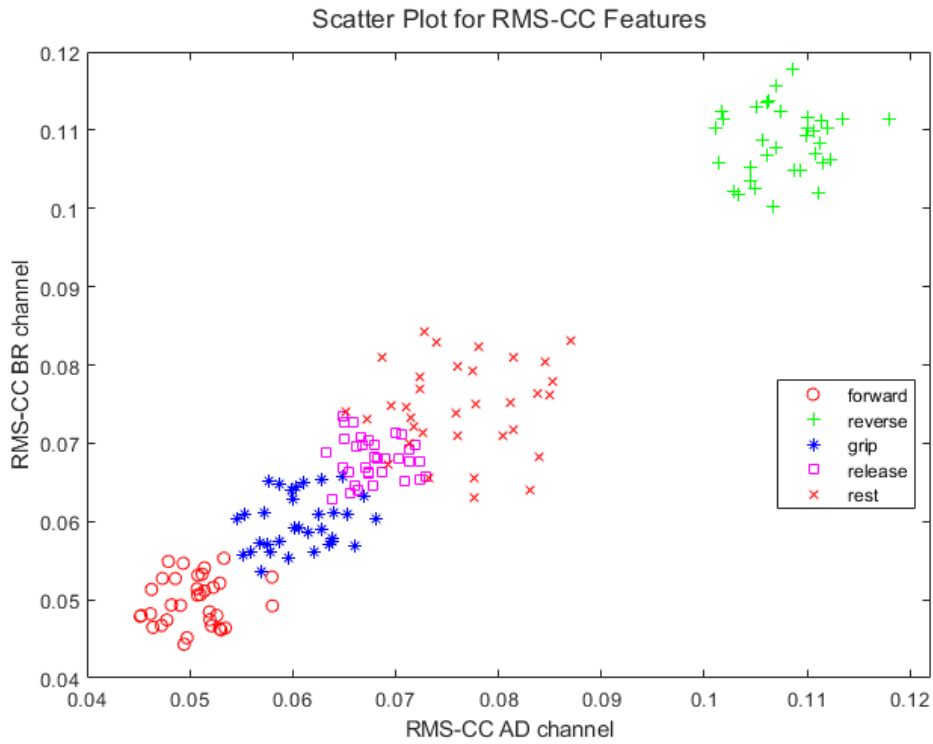


Fig 7.8: Scatter Plot for AD and BR channel

Implementing the proposed solution on the Embedded Platform generates the following classification accuracies as shown in Figure 7.9.

True Class	Reverse	1238	0	0	0	0	100%
	Forward	0	1182	56	0	0	95.48%
	RelDown	0	52	1170	16	0	94.51%
	GripUp	0	0	12	1163	53	93.94%
	Rest	0	0	0	54	1184	95.64%
Recall	100%	95.02%	94.51%	94.32%	95.72%		Precision
		Predicted Class					

Fig 7.9: Five-Way Confusion Matrix

It can be seen that the 'Reverse' class has a 100% classification accuracy as its feature space is clearly distinct from the other features. For the other classes there is a small overlap in the feature space which leads to some loss. The overall classification accuracy for the whole system is 95.31%.

7.7 Discussion

Designing a system that encapsulates all the elements of a complete solution onto an embedded platform has many challenges. Firstly, the proposed solution has to be tried and tested on a PC-based platform. Through research and experimentation, the final set of feature selection and machine learning methods are decided. The machine learning model is then trained and saved on the PC. To port this model over to the microcontroller, specialized software tools provided by the manufacturer have to be used. Using these tools, we can implement the trained model directly onto the hardware blocks within the microcontroller. Once that is done, we need to implement the signal processing and machine learning algorithms on the embedded board to complete the whole design.

Dealing with real-time data requires a lot of computing resources as several signal processing and feature extraction methods need to be executed before the machine learning model is used for the final classification. This can be achieved through the use of a multi-threading framework provided through a Real-Time Operating System. Several tasks can be performed concurrently allowing real-time classification of the signals with minimum delay.

The solution discussed in this paper has proposed a novel signal processing and feature extraction technique through the use of Cepstral features. This allows a good clean separation of the feature set with minimal overlap, leading to high classification accuracies. The computational requirement does not delay the system due to the use of the RTOS. Utilizing the proposed solution, we are able to achieve high classification accuracies on the embedded platform.

Designing a system that encapsulates all the elements of a complete solution onto an embedded platform has many challenges. Firstly, the proposed solution has to be tried and tested on a PC-based platform. Through research and experimentation, the final set of feature

selection and machine learning methods are decided. The machine learning model is then trained and saved on the PC. To port this model over to the microcontroller, specialized software tools provided by the manufacturer have to be used. Using these tools, we can implement the trained model directly onto the hardware blocks within the microcontroller. Once that is done, we need to implement the signal processing and machine learning algorithms on the embedded board to complete the whole design.

Dealing with real-time data requires a lot of computing resources as several signal processing and feature extraction methods need to be executed before the machine learning model is used for the final classification. This can be achieved through the use of a multi-threading framework provided through a Real-Time Operating System. Several tasks can be performed concurrently allowing real-time classification of the signals with minimum delay. The solution discussed in this paper has proposed a novel signal processing and feature extraction technique through the use of Cepstral features. This allows a good clean separation of the feature set with minimal overlap, leading to high classification accuracies. The computational requirement does not delay the system due to the use of the RTOS. Utilizing the proposed solution, we are able to achieve high classification accuracies on the embedded platform.

With these developments, we will be able to develop rehabilitative robotic systems that are able to be fully deployed on the edge without the need for any network connectivity. It will benefit many users in the activities-for-daily-living.

Chapter 8

Conclusion and Future Work

The time-series nature of physiological signals like EEG and EMG require the use of novel techniques and architectures to extract meaningful and accurate information from them.

The Novel EEG Model proposed in Chapter 3 introduced the Gamma-Peak feature. This was able to accurately determine the temporal vicinity of the motor movement based on a single channel data. The 'Slope & Peak Detection' algorithm together with the ERD/ERS features provided high levels of accuracy in the range of 89.58% to 92.29%. Using a single channel, these results exceeded other methods mentioned in the Literature.

In Chapter 4, we introduced the Novel EMG model. Other techniques in the Literature used a fixed set of sensor data in order to classify different movements. In the proposed approach, we used different combinations of sensor data for different movements. The proposed approach used a combination of features such as Root-Mean-Square (RMS), Discrete Wavelet Transform (DWT), and Entropy to generate the feature set. Together with the Feedforward Neural Network, it was able to accurately classify four different movements with an accuracy level of up to 93.5%.

Chapter 5 introduced the integration of Fuzzy Logic together with the Long Short-Term Memory (LSTM) network. Physiological signals exhibit both time and frequency changes in specific electrode regions associated with the type of activity or stimulus. To fully understand the changes in these signals, we need to be able to analyze its change over a subset of time. This requires a solution that has the ability to remember and detect specific patterns in the signals over time.

The Long Short-Term Memory (LSTM) is a network that is able to achieve this objective. LSTM

has also been utilized in many other applications that require memory and the “context” of the data with respect to its past. As the data is inherently embedded in the signals of different sensors, the use of Fuzzy Logic allows to naturally integrate this information together to get a better context of the data. The use of LSTM+Fuzzy Logic demonstrated a significant improvement in the classification accuracy for both EEG and EMG signals. A comparison study across different Deep Learning Techniques demonstrated the superiority of the proposed method.

A novel BIO-Inspired Fuzzy Inference System (BIOFIS) was proposed in Chapter 6. Using this model, we are able to capture the temporal intra-channel relationship together with the inter-channel relationship. Together with an improved self-clustering fuzzy rule-base, we are able to generate higher classification accuracies.

Compared to the mentioned methods in the Literature, the proposed technique in this paper uses an approach that considers the inter-channel and intra-channel relationships between various sensors. Most of the current research in this area is focused on small movements that isolate a body part, like the opening or closing of the hand. In this research, what we have demonstrated is the ability of the proposed system to accurately decode and classify actions that are full-range in nature. These actions represent real-world movements used in ADL.

The proposed methodology fuses multiple signals together and selects that most prominent ones for the classification, instead of relying of predefined signals. This is significant as in a real-world scenario, the user may have certain physiological challenges that may impair the quality of data captured from certain sensors. BIOFIS adapts to the available signals and is still able to generate high classification accuracies compared to the state-of-the-art.

The achieved accuracy of up to 98% for the 4-class classification is a good result the demonstrates the effectiveness of the BIOFIS model.

To come full-circle, in Chapter 7, we implemented signal processing and machine learning techniques on an advanced ARM-based microcontroller platform to perform edge computing. Using this microcontroller, together with a real-time operating system to achieve concurrency, we were able to achieve a high-level of accuracy on an embedded platform with limited computing power and resources.

In conclusion, rehabilitative robotics holds great promise for improving the quality of life for individuals with motor impairments. The integration of signal processing and machine learning algorithms into robotic systems offers the potential for more effective, efficient, and personalized rehabilitation therapies. This thesis has significantly contributed to the field by investigating and advancing the use of signal processing and machine learning algorithms in rehabilitative robotics. Through the exploration of various techniques, this research seeks to enhance the rehabilitation process and ultimately improve the recovery outcomes and quality of life for individuals who need it.

The future direction of this research should focus on integrating more advanced machine learning algorithms onto embedded platforms. This will allow the proposed models to be further enhanced with improved accuracy. Transformer models are gaining widespread popularity at the moment. With further improvements in the model, in the areas of memory footprint and power consumption, transformer models may prove to be very promising in embedded platforms that perform edge computing. Another direction would be the development of complete rehabilitative robotic systems that integrate the models that have been developed in this thesis. This will allow further improvements in the overall system in signal processing and machine learning techniques for a wide range of sensors and robotic platforms.

REFERENCES

1. Rothwell, P.M., P. Algra A Fau - Amarenco, and P. Amarenco, *Medical treatment in acute and long-term secondary prevention after transient ischaemic attack and ischaemic stroke*. (1474-547X (Electronic)).
2. Reinkensmeyer, D.J., et al., *Design of robot assistance for arm movement therapy following stroke*. *Advanced Robotics*, 2001. **14**(7): p. 625-637.
3. Westlake, K.P. and N.N. Byl, *Neural plasticity and implications for hand rehabilitation after neurological insult*. (1545-004X (Electronic)).
4. Hwang, C.H., D.-S. Seong Jw Fau - Son, and D.S. Son, *Individual finger synchronized robot-assisted hand rehabilitation in subacute to chronic stroke: a prospective randomized clinical trial of efficacy*. (1477-0873 (Electronic)).
5. Barbeau, H., *Locomotor training in neurorehabilitation: emerging rehabilitation concepts*. (1545-9683 (Print)).
6. Brewer, B.R., L.C. McDowell Sk Fau - Worthen-Chaudhari, and L.C. Worthen-Chaudhari, *Poststroke upper extremity rehabilitation: a review of robotic systems and clinical results*. (1074-9357 (Print)).
7. Hallett, M., *Recent Advances in Stroke Rehabilitation*. *Neurorehabilitation & Neural Repair*, 2002. **16**: p. 211-217.
8. Hsieh, Y.W., et al., *Predicting clinically significant changes in motor and functional outcomes after robot-assisted stroke rehabilitation*. (1532-821X (Electronic)).
9. Masia, L., et al., *Design and Characterization of Hand Module for Whole-Arm Rehabilitation Following Stroke*. *IEEE/ASME Transactions on Mechatronics*, 2007. **12**(4): p. 399-407.
10. Krebs, H.I., et al., *A paradigm shift for rehabilitation robotics*. *IEEE Engineering in Medicine and Biology Magazine*, 2008. **27**(4): p. 61-70.
11. Lum, P.S., et al. *The MIME robotic system for upper-limb neuro-rehabilitation: results from a clinical trial in subacute stroke*. in *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005*. 2005.
12. Schabowsky, C.N., et al., *Development and pilot testing of HEXORR: hand EXOskeleton rehabilitation robot*. *J Neuroeng Rehabil*, 2010. **7**: p. 36.
13. Sale, P., M. Lombardi V Fau - Franceschini, and M. Franceschini, *Hand robotics rehabilitation: feasibility and preliminary results of a robotic treatment in patients with hemiparesis*. (2042-0056 (Electronic)).
14. Cheng, X., et al., *Design of an Upper Limb Rehabilitation Robot Based on Medical Theory*. *Procedia Engineering*, 2011. **15**: p. 688-692.
15. Emken, J.L., R. Benitez, and D.J. Reinkensmeyer, *Human-robot cooperative movement training: Learning a novel sensory motor transformation during walking with robotic assistance-as-needed*. *Journal of NeuroEngineering and Rehabilitation*, 2007. **4**(1): p. 8.
16. Daly, J.J. and R.L. Ruff, *Construction of efficacious gait and upper limb functional interventions based on brain plasticity evidence and model-based measures for stroke patients*. (1537-744X (Electronic)).
17. Nasios G, D.E., Messinis L., *From Broca and Wernicke to the Neuromodulation Era: Insights of Brain Language Networks for Neurorehabilitation*. *Behav Neurol*, 2019.
18. Saito, S., et al. *Basic study of brain injury mechanism*. in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*. 2010.
19. Jarosz, Q., *Neuron*: https://en.wikipedia.org/wiki/File:Neuron_Hand-tuned.svg.
20. *Stroke*: <https://www.stroke.org/en/about-stroke>.
21. Riddoch, M.J., G.W. Humphreys, and A. Bateman, *Stroke*. *Physiotherapy*, 1995. **81**(11): p. 689-694.

22. Scrivener, K., C. Sherrington, and K. Schurr, *A systematic review of the responsiveness of lower limb physical performance measures in inpatient care after stroke*. BMC Neurology, 2013. **13**: p. 4.
23. Stewart, K.C., J.H. Cauraugh, and J.J. Summers,, *Bilateral movement training and stroke rehabilitation: A systematic review and meta-analysis*. Journal of the Neurological Sciences, 2006. **244**(1): p. 89-95.
24. Butt, M., et al. *Assessment of Neuroplasticity Using EEG Signal in Rehabilitation of Brain Stem Stroke Patients*. in *2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. 2020.
25. Gayla L. Poling, J.M.W., Thomas Kerwin., *The Role of Multisensory Feedback in Haptic Surface Perception*. 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS'03). 2003.
26. Broetz, D., et al., *Combination of brain-computer interface training and goaldirected physical therapy in chronic stroke: a case report*. Neurorehabilitation and Neural Repair, 2010. **24**(7): p. 9.
27. Cameirao, M.S., et al., *Neurorehabilitation using the virtual reality based Rehabilitation Gaming System: methodology, design, psychometrics, usability and validation*. Journal of NeuroEngineering and Rehabilitation, 2010. **7**: p. 48.
28. Cesqui, B., et al., *EMG-based pattern recognition approach in post stroke robot-aided rehabilitation: a feasibility study*. Journal of NeuroEngineering and Rehabilitation, 2013. **10**(1): p. 75.
29. Lam, P., et al., *A haptic-robotic platform for upper-limb reaching stroke therapy: preliminary design and evaluation results*. Journal of NeuroEngineering and Rehabilitation, 2008. **5**: p. 15.
30. Wang, L. and J. Orchard, *Investigating the Evolution of a Neuroplasticity Network for Learning*. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2019. **49**(10): p. 2131-2143.
31. Rumelhart, D.E., G.E. Hinton, and R.J. Williams, *Learning representations by back-propagating errors*. Nature, 1986. **323**: p. 533-536.
32. Bengio, Y., et al., *Towards Biologically Plausible Deep Learning*. arXiv e-prints, 2015: p. arXiv:1502.04156.
33. D. G. Stork, *Is backpropagation biologically plausible?* Joint Conf. Neural Netw. (IJCNN), 1989. **2**: p. 241-246.
34. He, W., et al., *Multiagent systems on multilayer networks*. Synchronization analysis and network design, 2017. **47**(7): p. 1655-1667.
35. C. L. P. Chen, J.W., C.-H. Wang, and L. Chen., *A new learning algorithm for a fully connected neuro-fuzzy inference system*. IEEE Trans. Neural Netw. Learn. Syst., 2014. **25**: p. 1741-1757.
36. Hinton, G.E., S. Osindero, and Y.W. Teh, *A fast learning algorithm for deep belief nets*. Neural Comput, 2006. **18**(7): p. 1527-54.
37. Janmajaya, M., et al., *Interval Type-2 Fuzzy Restricted Boltzmann Machine for the Enhancement of Deep Learning*, in *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. 2019, IEEE Press: New Orleans, LA, USA. p. 1–6.
38. Chen, C.L.P. and Z. Liu, *Broad Learning System: An Effective and Efficient Incremental Learning System Without the Need for Deep Architecture*. IEEE Trans Neural Netw Learn Syst, 2018. **29**(1): p. 10-24.
39. Hebb, D.O., *The Organization of Behavior*. 1949.
40. Lynch, T.D.a.G., *Long-term potentiation and depression of synaptic responses in the rat hippocampus: localization and frequency dependency*. J. Physiol., 1978: p. 353-367.
41. N. Rochester, J.H., L. Haibt and W. Duda, *Tests on a cell assembly theory of the action of the brain, using a large scale digital computer*. IRE Trans. Inf. Theory, 1956: p. 80-93.

42. Bienenstock, E.L., L.N. Cooper, and P.W. Munro, *Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex*. The Journal of Neuroscience, 1982. **2**(1): p. 32.
43. Dobkin, B.H., *Strategies for stroke rehabilitation*. Lancet Neurol, 2004. **3**(9): p. 528-36.
44. Takeuchi, N. and S.-I. Izumi, *Maladaptive Plasticity for Motor Recovery after Stroke: Mechanisms and Approaches*. Neural Plasticity, 2012. **2012**: p. 359728.
45. Belda-Lois, J.M., et al., *Rehabilitation of gait after stroke: a review towards a top-down approach*. J Neuroeng Rehabil, 2011. **8**: p. 66.
46. Thrasher, T.A., et al., *Rehabilitation of reaching and grasping function in severe hemiplegic patients using functional electrical stimulation therapy*. Neurorehabil Neural Repair, 2008. **22**(6): p. 706-14.
47. Hubbard, I.J., et al., *Task-specific training: evidence for and translation to clinical practice*. Occup Ther Int, 2009. **16**(3-4): p. 175-89.
48. Zimmerli, L., et al., *Validation of a mechanism to balance exercise difficulty in robot-assisted upper-extremity rehabilitation after stroke*. Journal of NeuroEngineering and Rehabilitation, 2012. **9**(1): p. 6.
49. Thielman, G.T., C.M. Dean, and A.M. Gentile, *Rehabilitation of reaching after stroke: task-related training versus progressive resistive exercise*. Arch Phys Med Rehabil, 2004. **85**(10): p. 1613-8.
50. Lang, C.E., et al., *Measurement of Upper-Extremity Function Early After Stroke: Properties of the Action Research Arm Test*. Archives of Physical Medicine and Rehabilitation, 2006. **87**(12): p. 1605-1610.
51. Page, S.J., G.D. Fulk, and P. Boyne, *Clinically important differences for the upper-extremity Fugl-Meyer Scale in people with minimal to moderate impairment due to chronic stroke*. Phys Ther, 2012. **92**(6): p. 791-8.
52. Ferraro, M., et al., *Assessing the motor status score: a scale for the evaluation of upper limb motor outcomes in patients after stroke*. Neurorehabil Neural Repair, 2002. **16**(3): p. 283-9.
53. Wolf, S.L., et al., *Assessing Wolf motor function test as outcome measure for research in patients after stroke*. Stroke, 2001. **32**(7): p. 1635-9.
54. Li, K.-y., et al., *Population based norms for the box and blocks test in healthy right-handed Taiwanese adults*. Biomedical Journal, 2020. **43**(6): p. 484-489.
55. Reedman, S.E., et al., *The Jebsen Taylor Test of Hand Function: A Pilot Test-Retest Reliability Study in Typically Developing Children*. Phys Occup Ther Pediatr, 2016. **36**(3): p. 292-304.
56. Feys, P., et al., *The Nine-Hole Peg Test as a manual dexterity performance measure for multiple sclerosis*. Multiple sclerosis (Houndmills, Basingstoke, England), 2017. **23**(5): p. 711-720.
57. Lo, A.C., et al., *Robot-Assisted Therapy for Long-Term Upper-Limb Impairment after Stroke*. New England Journal of Medicine, 2010. **362**(19): p. 1772-1783.
58. Mehrholz, J., et al., *Electromechanical and robot-assisted arm training for improving generic activities of daily living, arm function, and arm muscle strength after stroke*. Cochrane Database Syst Rev, 2012(6): p. CD006876.
59. Turk, R., et al., *Therapeutic effectiveness of electric stimulation of the upper-limb poststroke using implanted microstimulators*. Arch Phys Med Rehabil, 2008. **89**(10): p. 1913-22.
60. Qian, Z. and Z. Bi, *Recent Development of Rehabilitation Robots*. Advances in Mechanical Engineering, 2014. **7**(2): p. 563062.
61. Johnson, M.J., et al., *Potential of a suite of robot/computer-assisted motivating systems for personalized, home-based, stroke rehabilitation*. Journal of NeuroEngineering and Rehabilitation, 2007. **4**(1): p. 6.
62. Mokhtari M, F.M., Abdulrazak B, et al. , *3 Toward a human-friendly user interface to control an assistive robot in the context of smart homes*. . Adv Rehabil Robot, 2006(306): p. 47-56.

63. Bi, Z.M., Y. Lin, and W.J. Zhang, *The general architecture of adaptive robotic systems for manufacturing applications*. Robotics and Computer-Integrated Manufacturing, 2010. **26**(5): p. 461-470.
64. Suppiah, R., et al. *An Electromyography-aided Robotics Hand for Rehabilitation – A Proof-of-Concept Study*. in *2020 IEEE REGION 10 CONFERENCE (TENCON)*. 2020.
65. Krebs, H.I., et al., *Rehabilitation robotics: pilot trial of a spatial extension for MIT-Manus*. Journal of NeuroEngineering and Rehabilitation, 2004. **1**(1): p. 5.
66. Lum, P.S., et al., *Robot-assisted movement training compared with conventional therapy techniques for the rehabilitation of upper-limb motor function after stroke*. (0003-9993 (Print)).
67. Reinkensmeyer, D.J., et al., *Understanding and treating arm movement impairment after chronic brain injury: progress with the ARM guide*. (0748-7711 (Print)).
68. Staubli, P., et al., *Effects of intensive arm training with the rehabilitation robot ARMin II in chronic stroke patients: four single-cases*. Journal of NeuroEngineering and Rehabilitation, 2009. **6**(1): p. 46.
69. van der Linde, R.Q. and P. Lammertse, *HapticMaster – a generic force controlled robot for human interaction*. Industrial Robot: An International Journal, 2003. **30**(6): p. 515-524.
70. Rosati, G., S. Gallina P Fau - Masiero, and S. Masiero, *Design, implementation and clinical tests of a wire-based robot for neurorehabilitation*. (1534-4320 (Print)).
71. Fazekas, G., A. Horvath M Fau - Toth, and A. Toth, *A novel robot training system designed to supplement upper limb physiotherapy of patients with spastic hemiparesis*. Int J Rehabil Res., 2006. **29**(0342-5282).
72. Qassim, H.M. and W.Z. Wan Hasan, *A Review on Upper Limb Rehabilitation Robots*. Applied Sciences, 2020. **10**(19).
73. NeuroRehab. Available from: <https://www.neurorehabdirectory.com/rehab-products/inmotion/>.
74. Harwin, W., et al., *The GENTLE/S project: A new method of delivering neuro-rehabilitation*. Assistive technology—added value to the quality of life AAATE, 2001. **1**: p. 36-41.
75. Frisoli, A., et al., *Robotic assisted rehabilitation in Virtual Reality with the L-EXOS*. (0926-9630 (Print)).
76. Dehem, S., et al., *Development of a robotic upper limb assessment to configure a serious game*. NeuroRehabilitation, 2019. **44**: p. 263-274.
77. Stein, J., et al., *Robot-assisted exercise for hand weakness after stroke: a pilot study*. (1537-7385 (Electronic)).
78. Adamovich, S.V., et al., *Design of a complex virtual reality simulation to train finger motion for persons with hemiparesis: a proof of concept study*. (1743-0003 (Electronic)).
79. Masia, L., et al. *Design, Characterization, and Impedance Limits of a Hand Robot*. in *2007 IEEE 10th International Conference on Rehabilitation Robotics*. 2007.
80. Bouzit, M., et al. *The Rutgers Master II-ND force feedback glove*. in *Proceedings 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. HAPTICS 2002*. 2002.
81. Dovat, L., et al., *HandCARE: a cable-actuated rehabilitation system to train hand function after stroke*. (1558-0210 (Electronic)).
82. Einav, O., et al., *Development and validation of the first robotic scale for the clinical assessment of upper extremity motor impairments in stroke patients*. (1074-9357 (Print)).
83. Giralt X., A.L., Casals A., Amat J. (2013) . In: Pons J., Torricelli D., Pajaro M. (eds) *Robotic Platform to Evaluate the Assistance and Assessment on the Rehabilitation Loop*. Converging Clinical and Engineering Research on Neurorehabilitation. Biosystems & Biorobotics, 2013. **1**.
84. Wolbrecht, E.T., D.J. Reinkensmeyer, and J.E. Bobrow, *Pneumatic Control of Robots for Rehabilitation*. The International Journal of Robotics Research, 2009. **29**(1): p. 23-38.

85. Rehmat, N., et al., *Upper limb rehabilitation using robotic exoskeleton systems: a systematic review*. International Journal of Intelligent Robotics and Applications, 2018. **2**.
86. Marchal-Crespo, L. and D.J. Reinkensmeyer, *Review of control strategies for robotic movement training after neurologic injury*. Journal of NeuroEngineering and Rehabilitation, 2009. **6**(1): p. 20.
87. Kahn, L.E., et al., *Robot-assisted reaching exercise promotes arm movement recovery in chronic hemiparetic stroke: a randomized controlled pilot study*. Journal of neuroengineering and rehabilitation, 2006. **3**: p. 12-12.
88. Mazzoleni, S., et al., *Effects of proximal and distal robot-assisted upper limb rehabilitation on chronic stroke recovery*. (1878-6448 (Electronic)).
89. Prange, G.B., et al., *Systematic review of the effect of robot-aided therapy on recovery of the hemiparetic arm after stroke*. (1938-1352 (Electronic)).
90. Fasoli, S.E., et al., *Effects of robotic therapy on motor impairment and recovery in chronic stroke*. (0003-9993 (Print)).
91. Zhang, M., T.C. Davies, and S. Xie, *Effectiveness of robot-assisted therapy on ankle rehabilitation – a systematic review*. Journal of NeuroEngineering and Rehabilitation, 2013. **10**(1): p. 30.
92. Pachalska, M., et al., *Rehabilitation of an artist after right-hemisphere stroke*. (1643-3750 (Electronic)).
93. Hill, A.V., *The heat of shortening and the dynamic constants of muscle*. . Proc. R. Soc. Lond. Ser. B Biol. Sci., 1938. **126**: p. 136-195.
94. Rosen, J., et al., *A myosignal-based powered exoskeleton system*. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 2001. **31**: p. 210-222.
95. Behrens, M., et al., *Plyometric training improves voluntary activation and strength during isometric, concentric and eccentric contractions*. (1878-1861 (Electronic)).
96. Al-Mulla, M.R., M. Sepulveda F Fau - Colley, and M. Colley, *A review of non-invasive techniques to detect and predict localised muscle fatigue*. (1424-8220 (Electronic)).
97. Goslow Ge Jr Fau - Reinking, R.M., D.G. Reinking Rm Fau - Stuart, and D.G. Stuart, *The cat step cycle: hind limb joint angles and muscle lengths during unrestrained locomotion*. (0362-2525 (Print)).
98. Hoffer Ja Fau - Caputi, A.A., et al., *Roles of muscle activity and load on the relationship between muscle spindle length and whole muscle length in the freely walking cat*. (0079-6123 (Print)).
99. Subasi, A., *Classification of EMG signals using PSO optimized SVM for diagnosis of neuromuscular disorders*. (1879-0534 (Electronic)).
100. Rogers, D.R. and D.T. Maclsaac, *A comparison of EMG-based muscle fatigue assessments during dynamic contractions*. (1873-5711 (Electronic)).
101. Rechy-Ramirez, E.J. and H. Hu, *Bio-signal based control in assistive robots: a survey*. Digital Communications and Networks, 2015. **1**(2): p. 85-101.
102. Yan, T., et al., *Review of assistive strategies in powered lower-limb orthoses and exoskeletons*. Robotics and Autonomous Systems, 2015. **64**: p. 120-136.
103. Lorrain, T., N. Jiang, and D. Farina, *Influence of the training set on the accuracy of surface EMG classification in dynamic contractions for the control of multifunction prostheses*. Journal of NeuroEngineering and Rehabilitation, 2011. **8**(1): p. 25.
104. Phinyomark, A., et al., *EMG feature evaluation for improving myoelectric pattern recognition robustness*. Expert Systems with Applications, 2013. **40**(12): p. 4832-4840.
105. Phinyomark, A., P. Phukpattaranont, and C. Limsakul, *Feature reduction and selection for EMG signal classification*. Expert Systems with Applications, 2012. **39**(8): p. 7420-7431.
106. Tsai, A.-C., et al., *A comparison of upper-limb motion pattern recognition using EMG signals during dynamic and isometric muscle contractions*. Biomedical Signal Processing and Control, 2014. **11**: p. 17-26.

107. Raez, M.B.I., M.S. Hussain, and F. Mohd-Yasin, *Techniques of EMG signal analysis: detection, processing, classification and applications*. Biological procedures online, 2006. **8**: p. 11-35.
108. Englehart, K. and B. Hudgins, *A robust, real-time control scheme for multifunction myoelectric control*. IEEE Transactions on Biomedical Engineering, 2003. **50**(7): p. 848-854.
109. Farina, D. and R. Merletti, *Comparison of algorithms for estimation of EMG variables during voluntary isometric contractions*. Journal of Electromyography and Kinesiology, 2000. **10**(5): p. 337-349.
110. Matsubara, T. and J. Morimoto, *Bilinear modeling of EMG signals to extract user-independent features for multiuser myoelectric interface*. (1558-2531 (Electronic)).
111. Ahmad, S., *Moving approximate entropy and its application to the electromyographic control of an artificial hand*. 2009.
112. Englehart, K., B. Hudgins, and P. Parker, *A wavelet-based continuous classification scheme for multifunction myoelectric control*. IEEE transactions on bio-medical engineering, 2001. **48**: p. 302-11.
113. Balbinot, A. and G. Favieiro, *A neuro-fuzzy system for characterization of arm movements*. Sensors (Basel, Switzerland), 2013. **13**(2): p. 2613-2630.
114. Bin Ahmad Nadzri, A.A., et al., *Characterization of surface electromyography using time domain features for determining hand motion and stages of contraction*. (1879-5447 (Electronic)).
115. Kendell, C., et al., *A novel approach to surface electromyography: an exploratory study of electrode-pair selection based on signal characteristics*. (1743-0003 (Electronic)).
116. Al-Angari, H.M., et al., *Distance and mutual information methods for EMG feature and channel subset selection for classification of hand movements*. Biomedical Signal Processing and Control, 2016. **27**: p. 24-31.
117. Chowdhury, R.H., et al., *Surface electromyography signal processing and classification techniques*. (1424-8220 (Electronic)).
118. Di Nardo, F., et al., *Assessment of the ankle muscle co-contraction during normal gait: a surface electromyography study*. (1873-5711 (Electronic)).
119. Abbaspour, S. and A. Fallah, *Removing ECG Artifact from the Surface EMG Signal Using Adaptive Subtraction Technique*. Journal of biomedical physics & engineering, 2014. **4**(1): p. 33-38.
120. Lee, J.M. and C.H. Kim. *Detection of Gait Intention with an Insole Device*. in *2019 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. 2019.
121. Ludwig, K.A., et al., *Using a Common Average Reference to Improve Cortical Neuron Recordings From Microelectrode Arrays*. Journal of Neurophysiology, 2009. **101**(3): p. 1679-1689.
122. Phinyomark, A., et al., *The usefulness of mean and median frequencies in electromyography analysis*. Computational intelligence in electromyography analysis-A perspective on current applications and future challenges, 2012: p. 195-220.
123. Syam, S.H.F., et al. *Comparing common average referencing to laplacian referencing in detecting imagination and intention of movement for brain computer interface*.
124. Kleine, B.U., et al., *Using two-dimensional spatial information in decomposition of surface EMG signals*. Journal of Electromyography and Kinesiology, 2007. **17**(5): p. 535-548.
125. Hudgins, B., P. Parker, and R.N. Scott, *A new strategy for multifunction myoelectric control*. IEEE Transactions on Biomedical Engineering, 1993. **40**(1): p. 82-94.
126. Wan Daud, W.M.B., et al., *Features Extraction of Electromyography Signals in Time Domain on Biceps Brachii Muscle*. International Journal of Modeling and Optimization, 2013. **3**: p. 515-519.
127. Ahsan, M.R., M. Ibrahimy, and O. Khalifa, *Neural Network Classifier for Hand Motion Detection from EMG Signal*. 2011. p. 536-541.

128. Sun R Fau - Song, R., K.-y. Song R Fau - Tong, and K.Y. Tong, *Complexity analysis of EMG signals for patients after stroke during robot-aided rehabilitation training using fuzzy approximate entropy*. (1558-0210 (Electronic)).
129. Asghari Oskoei, M. and H. Hu, *Myoelectric control systems—A survey*. Biomedical Signal Processing and Control, 2007. **2**(4): p. 275-294.
130. Bariouli, R., et al. *Evaluation of EMG Signal Time Domain Features for Hand Gesture Distinction*. in *2019 16th International Multi-Conference on Systems, Signals & Devices (SSD)*. 2019.
131. Khushaba, R.N., A. Al-Ani, and A. Al-Jumaily, *Orthogonal Fuzzy Neighborhood Discriminant Analysis for Multifunction Myoelectric Hand Control*. IEEE Transactions on Biomedical Engineering, 2010. **57**(6): p. 1410-1419.
132. Tkach, D., H. Huang, and T.A. Kuiken, *Study of stability of time-domain features for electromyographic pattern recognition*. Journal of NeuroEngineering and Rehabilitation, 2010. **7**(1): p. 21.
133. Oskoei, M.A. and H. Hu, *Support Vector Machine-Based Classification Scheme for Myoelectric Control Applied to Upper Limb*. IEEE Transactions on Biomedical Engineering, 2008. **55**(8): p. 1956-1965.
134. Englehart, K., et al., *Classification of the myoelectric signal using time-frequency based representations*. (1350-4533 (Print)).
135. Basu, B., S. Nagarajaiah, and A. Chakraborty, *Online Identification of Linear Time-varying Stiffness of Structural Systems by Wavelet Analysis*. Structural Health Monitoring-an International Journal - STRUCT HEALTH MONIT, 2008. **7**.
136. Nirenberg Lm Fau - Hanley, J., E.B. Hanley J Fau - Stear, and E.B. Stear, *A new approach to prosthetic control: EEG motor signal tracking with an adaptively designed phase-locked loop*. IEEE Trans Biomed Eng. **18**(0018-9294).
137. Nicolas-Alonso, L.F. and J. Gomez-Gil, *Brain computer interfaces, a review*. Sensors (Basel), 2012. **12**(1424-8220): p. 1211-1279.
138. Lebedev, M.A. and M.A. Nicolelis, *Brain-Machine Interfaces: From Basic Science to Neuroprostheses and Neurorehabilitation*. Physiol Rev., 2017. **97**(1522-1210).
139. Birbaumer, N. and L.G. Cohen, *Brain-computer interfaces: communication and restoration of movement in paralysis*. J Physiol., 2007. **579**(0022-3751).
140. Machado, S., et al., *EEG-based brain-computer interfaces: an overview of basic concepts and clinical applications in neurorehabilitation*. Reviews in the Neurosciences, 2010. **21**(0334-1763).
141. Moghimi, S., et al., *A review of EEG-based brain-computer interfaces as access pathways for individuals with severe disabilities*. Assist Technol., 2013. **25**(1040-0435).
142. Birbaumer, N., *Breaking the silence: Brain-computer interfaces (BCI) for communication and motor control*. Psychophysiology, 2006. **43**: p. 517-32.
143. Meng, J., et al., *Noninvasive Electroencephalogram Based Control of a Robotic Arm for Reach and Grasp Tasks*. Scientific Reports, 2016. **6**(1): p. 38565.
144. Daly, J.J. and J.R. Wolpaw, *Brain-computer interfaces in neurological rehabilitation*. Lancet Neurol, 2008(1474-4422).
145. Pfurtscheller, G. and C. Neuper, *Motor imagery activates primary sensorimotor area in humans*. (0304-3940 (Print)).
146. Morash, V., et al., *Classifying EEG signals preceding right hand, left hand, tongue, and right foot movements and motor imageries*. (1388-2457 (Print)).
147. Pfurtscheller, G. and F.H. Lopes da Silva, *Event-related EEG/MEG synchronization and desynchronization: basic principles*. Clin Neurophysiol., 1999. **110**(1388-2457).
148. Wolpaw, J.R. and D.J. McFarland, *Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans*. (0027-8424 (Print)).

149. Bhattacharyya, S., et al. *Performance analysis of left/right hand movement classification from EEG signal by intelligent algorithms*. in *2011 IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB)*. 2011.
150. Wolpaw, J.R., et al., *An EEG-based brain-computer interface for cursor control*. *Electroencephalography and Clinical Neurophysiology*, 1999. **78**(0013-4694).
151. Wolpaw, J.R. and D.J. McFarland, *Multichannel EEG-based brain-computer communication*. *Electroencephalography and Clinical Neurophysiology*, 1994. **90**(0013-4694).
152. Ramos-Murguialday, A., et al., *Brain-Computer Interface for a Prosthetic Hand Using Local Machine Control and Haptic Feedback*. 2007: IEEE 10th International Conference on Rehabilitation Robotics. 609-613.
153. Ang, K., et al., *Clinical study of neurorehabilitation in stroke using EEG-based motor imagery brain-computer interface with robotic feedback*. Vol. 2010. 2010. 5549-52.
154. Sarac M Fau - Koyas, E., et al., *Brain Computer Interface based robotic rehabilitation with online modification of task speed*. *IEEE Int Conf Rehabil Robot*, 2013(1945-7901).
155. Pereira, J., et al., *EEG neural correlates of goal-directed movement intention*. *NeuroImage*, 2017. **149**.
156. Hochberg, L.R., et al., *Neuronal ensemble control of prosthetic devices by a human with tetraplegia*. (1476-4687 (Electronic)).
157. Kim, S.-P., et al., *Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia*. *Journal of neural engineering*, 2008. **5**(4): p. 455-476.
158. Gu, Y., D. Dremstrup K Fau - Farina, and D. Farina, *Single-trial discrimination of type and speed of wrist movements from EEG recordings*. (1872-8952 (Electronic)).
159. Bradberry, T.J., J.L. Gentili Rj Fau - Contreras-Vidal, and J.L. Contreras-Vidal, *Fast attainment of computer cursor control with noninvasively acquired brain signals*. (1741-2552 (Electronic)).
160. Bradberry, T.J., R.J. Gentili, and J.L. Contreras-Vidal, *Reconstructing Three-Dimensional Hand Movements from Noninvasive Electroencephalographic Signals*. *The Journal of Neuroscience*, 2010. **30**(9): p. 3432.
161. Ferrez, P.W. and R.M.J. del, *Error-related EEG potentials generated during simulated brain-computer interaction*. (0018-9294 (Print)).
162. Luth, T., et al., *Low level control in a semi-autonomous rehabilitation robotic system via a Brain-Computer Interface*. 2007. 721-728.
163. Li, Y., et al., *An EEG-based BCI system for 2-D cursor control by combining Mu/Beta rhythm and P300 potential*. *IEEE Transactions on Biomedical Engineering*, 2010. **57**: p. 2495-505.
164. Duan, F., et al., *Design of a Multimodal EEG-based Hybrid BCI System with Visual Servo Module*. *IEEE Transactions on Autonomous Mental Development*, 2015. **7**: p. 1-1.
165. Kim, B.H., M. Kim, and S. Jo, *Quadcopter flight control using a low-cost hybrid interface with EEG-based classification and eye tracking*. (1879-0534 (Electronic)).
166. Kim M Fau - Kim, B.H., S. Kim Bh Fau - Jo, and S. Jo, *Quantitative evaluation of a low-cost noninvasive hybrid interface based on EEG and eye movement*. (1558-0210 (Electronic)).
167. Choi, I., et al., *A systematic review of hybrid brain-computer interfaces: Taxonomy and usability perspectives*. (1932-6203 (Electronic)).
168. Hong, K.-S. and M.J. Khan, *Hybrid Brain-Computer Interface Techniques for Improved Classification Accuracy and Increased Number of Commands: A Review*. *Frontiers in neurorobotics*, 2017. **11**: p. 35-35.
169. Banville, H. and T. Falk, *Recent advances and open challenges in hybrid brain-computer interfacing: a technological review of non-invasive human research*. *Brain-Computer Interfaces*, 2016. **3**: p. 1-38.
170. Rosenblatt, F., *Principles of neurodynamics. perceptrons and the theory of brain mechanisms*. 1961, Cornell Aeronautical Lab Inc Buffalo NY.

171. Cortes, C. and V. Vapnik, *Support-vector networks*. Machine Learning, 1995. **20**(3): p. 273-297.
172. Khushaba, R.N., Al-Jumaily, A., & Al-Ani, A., *Novel feature extraction method based on fuzzy entropy and wavelet packet transform for myoelectric control*. International Symposium on Communications and Information Technologies, 2007.
173. LeCun, Y., et al., *Backpropagation Applied to Handwritten Zip Code Recognition*. Neural Computation, 1989. **1**(4): p. 541-551.
174. Elman, J.L., *Finding structure in time*. Cognitive Science, 1990. **14**(2): p. 179-211.
175. Hochreiter, S. and J. Schmidhuber, *Long Short-Term Memory*. Neural Computation, 1997. **9**(8): p. 1735-1780.
176. Ballard, D.H. *Modular learning in neural networks*.
177. Ding, Z., et al., *sEMG-Based Gesture Recognition with Convolution Neural Networks*. Sustainability, 2018. **10**(6).
178. Shim, H.-m., et al. *EMG Pattern Classification by Split and Merge Deep Belief Network*. Symmetry, 2016. **8**, DOI: 10.3390/sym8120148.
179. Luo, R., et al., *A Low-Cost End-to-End sEMG-Based Gait Sub-Phase Recognition System*. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2020. **28**(1): p. 267-276.
180. Zhai, X., et al., *Self-Recalibrating Surface EMG Pattern Recognition for Neuroprosthesis Control Based on Convolutional Neural Network*. Frontiers in Neuroscience, 2017. **11**.
181. Haigh, K.Z., et al. *Machine Learning for Embedded Systems : A Case Study*. 2015.
182. Lee, J., et al. *Integrating machine learning in embedded sensor systems for Internet-of-Things applications*. in *2016 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. 2016.
183. Ronao, C.A. and S.-B. Cho, *Recognizing human activities from smartphone sensors using hierarchical continuous hidden Markov models*. International Journal of Distributed Sensor Networks, 2017. **13**.
184. Fafoutis, X., et al. *Extending the battery lifetime of wearable sensors with embedded machine learning*. in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*. 2018.
185. Langroudi, S.H.F., T. Pandit, and D. Kudithipudi. *Deep Learning Inference on Embedded Devices: Fixed-Point vs Posit*. in *2018 1st Workshop on Energy Efficient Machine Learning and Cognitive Computing for Embedded Applications (EMC2)*. 2018.
186. Olry, R., *Korbinian Brodmann (1868–1918)*. Journal of Neurology, 2010. **257**(12): p. 2112-2113.
187. Toga, A.W., et al., *Mapping the Human Connectome*. Neurosurgery, 2012. **71**(1): p. 1-5.
188. Luciw, M.D., E. Jarocka, and B.B. Edin, *Multi-channel EEG recordings during 3,936 grasp and lift trials with varying weight and friction*. Scientific Data, 2014. **1**(1): p. 140047.
189. Pfurtscheller, G., *Graphical display and statistical evaluation of event-related desynchronization (ERD)*. Electroencephalography and Clinical Neurophysiology, 1977. **43**(5): p. 757-760.
190. Pfurtscheller, G., *Event-related synchronization (ERS): an electrophysiological correlate of cortical areas at rest*. Electroencephalography and Clinical Neurophysiology, 1992. **83**(1): p. 62-69.
191. Wairagkar, M., Y. Hayashi, and S.J. Nasuto, *Dynamics of Long-Range Temporal Correlations in Broadband EEG During Different Motor Execution and Imagery Tasks*. (1662-4548 (Print)).
192. Schalk, G., et al., *BCI2000: a general-purpose brain-computer interface (BCI) system*. (0018-9294 (Print)).
193. Jeong, J.-H., et al., *Multimodal signal dataset for 11 intuitive movement tasks from single upper extremity during multiple recording sessions*. GigaScience, 2020. **9**(10).
194. Farrell, T.R. and R.F. Weir, *A comparison of the effects of electrode implantation and targeting on pattern classification accuracy for prosthesis control*. (1558-2531 (Electronic)).

195. Farina, R.M.a.D., *Surface Electromyography: Physiology, Engineering, and Applications*. Wiley, 2006.
196. Farina D Fau - Jiang, N., et al., *The extraction of neural information from the surface EMG for the control of upper-limb prostheses: emerging avenues and challenges*. (1558-0210 (Electronic)).
197. Guerrero, F.N., E.M. Spinelli, and M.A. Haberman, *Analysis and Simple Circuit Design of Double Differential EMG Active Electrode*. IEEE Transactions on Biomedical Circuits and Systems, 2016. **10**(3): p. 787-795.
198. Park, S.H. and S.P. Lee, *EMG pattern recognition based on artificial intelligence techniques*. (1063-6528 (Print)).
199. Liu, Y. and H. Huang. *Towards a high-stability EMG recognition system for prosthesis control: A one-class classification based non-target EMG pattern filtering scheme*. in *2009 IEEE International Conference on Systems, Man and Cybernetics*. 2009.
200. Li Z Fau - Wang, B., et al., *sEMG-based joint force control for an upper-limb power-assist exoskeleton robot*. (2168-2208 (Electronic)).
201. Mewett, D.T., H. Nazeran, and K.J. Reynolds. *Removing power line noise from recorded EMG*. in *2001 Conference Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 2001.
202. Mewett, D.T., H. Reynolds KJ Fau - Nazeran, and H. Nazeran, *Reducing power line interference in digitised electromyogram recordings by spectrum interpolation*. (0140-0118 (Print)).
203. Malboubi, M., et al. *Power line noise elimination from EMG signals using adaptive Laguerre filter with fuzzy step size*. in *2010 17th Iranian Conference of Biomedical Engineering (ICBME)*. 2010.
204. Liu, J. and P. Zhou, *A novel myoelectric pattern recognition strategy for hand function restoration after incomplete cervical spinal cord injury*. (1558-0210 (Electronic)).
205. Celadon N Fau - Dosen, S., et al., *Individual finger classification from surface EMG: Influence of electrode set*. (2694-0604 (Electronic)).
206. Ning Y Fau - Zhu, X., et al., *Surface EMG decomposition based on K-means clustering and convolution kernel compensation*. (2168-2208 (Electronic)).
207. Ison M Fau - Vujaklija, I., et al., *High-Density Electromyography and Motor Skill Learning for Robust Long-Term Control of a 7-DoF Robot Arm*. (1558-0210 (Electronic)).
208. Triwiyanto, T., I. Pawana, and M. Purnomo, *An Improved Performance of Deep Learning Based on Convolution Neural Network to Classify the Hand Motion by Evaluating Hyper Parameter*. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2020. **PP**: p. 1-1.
209. Tsinganos, P., et al., *Improved Gesture Recognition Based on sEMG Signals and TCN*. 2019.
210. Hu, Y., et al., *A novel attention-based hybrid CNN-RNN architecture for sEMG-based gesture recognition*. PLOS ONE, 2018. **13**: p. e0206049.
211. Zhang, D., et al., *EEG-based Intention Recognition from Spatio-Temporal Representations via Cascade and Parallel Convolutional Recurrent Neural Networks*. 2017.
212. Tsiouris, K., et al., *A Long Short-Term Memory deep learning network for the prediction of epileptic seizures using EEG signals*. Computers in Biology and Medicine, 2018. **99**.
213. Greff, K., et al., *LSTM: A Search Space Odyssey*. IEEE Transactions on Neural Networks and Learning Systems, 2017. **28**(10): p. 2222-2232.
214. AUCROC, https://link.springer.com/referenceworkentry/10.1007/978-1-4419-9863-7_209.
215. Krasoulis, A., et al., *Improved prosthetic hand control with concurrent use of myoelectric and inertial measurements*. Journal of NeuroEngineering and Rehabilitation, 2017. **14**(1): p. 71.
216. Atzori, M., et al., *Characterization of a Benchmark Database for Myoelectric Movement Classification*. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2015. **23**(1): p. 73-83.

217. Ravi Suppiah, N.K., Anurag Sharma, Khalid Abidi, *Motor State Classification based on Electromyography (EMG) Signals using Wavelet Entropy and Neural Networks*. International Conference on Computational Science and Computational Intelligence (CSCI), 2021.
218. Bogert, B.P. *The quefreny analysis of time series for echoes : cepstrum, pseudo-autocovariance, cross-cepstrum and saphe cracking*. 1963.
219. <https://www.freertos.org/>.