DEVELOPMENT OF POWERED ANKLE PROSTHESIS CLASSIFICATION ALGORITHMS FOR STEP-UP VS. STEP-DOWN MOVEMENTS USING MACHINE LEARNING FED EXCLUSIVELY BY SURFACE ELECTROMYOGRAPHY

by

Onur Sürhan

B.S., in Mechanical Engineering, Boğaziçi University, 2019

Submitted to the Institute of Biomedical Engineering
in partial fulfillment of the requirements
for the degree of
Master of Science
in
Biomedical Engineering

Boğaziçi University
2021

DEVELOPMENT OF POWERED ANKLE PROSTHESIS CLASSIFICATION ALGORITHMS FOR STEP-UP VS. STEP-DOWN MOVEMENTS USING MACHINE LEARNING FED EXCLUSIVELY BY SURFACE ELECTROMYOGRAPHY

APPROVED BY:

Prof. Dr. Can Ali Yücesoy .....
(Thesis Advisor)

Assoc. Prof. Dr. Esin Öztürk Işık .....

Assist. Prof. Dr. İpek Şen .....

DATE OF APPROVAL: 14 September 2021

ACKNOWLEDGMENTS

I would like to thank, First of all, to my whole family, especially my mother, father, and brother, who have supported me in every way throughout my education life. Also to my grandmother and grandfather, Naciye and Nazmi Yılmaz, to whom I dedicate this thesis and believe that they watch me from heaven and are proud of me.

To all my teachers, especially my thesis advisor, Prof. Dr. Can Yücesoy, who raised me and played an inspirational role by advising me to reach where I am today.

To all my friends who made me who I am and who influenced my life, whether I noticed it or not.

Finally, for whom I share the same fate and whom I am grateful to have in my life every day. To my one and only love, Gizem Günbal, for her support, courage, motivation, and everything.

ACADEMIC ETHICS AND INTEGRITY STATEMENT

I, Onur Sürhan, hereby certify that I am aware of the Academic Ethics and Integrity Policy issued by the Council of Higher Education (YÖK) and I fully acknowledge all the consequences due to its violation by plagiarism or any other way.

Name :

Onur Sürhan

Signature:

Date:

14 / 09 / 2021

ABSTRACT

DEVELOPMENT OF POWERED ANKLE PROSTHESIS CLASSIFICATION ALGORITHMS FOR STEP-UP VS. STEP-DOWN MOVEMENTS USING MACHINE LEARNING FED EXCLUSIVELY BY SURFACE ELECTROMYOGRAPHY

Rapidly increasing developments in the field of artificial intelligence yield implications for motion supportive devices. Powered ankle prostheses (PAP) employ actuation mechanisms that aim to support amputees during locomotion. Surface electromyography (sEMG) allows muscles contribution in developing such algorithms and their usage as the exclusive sensing source is novel. Different techniques of processing raw sEMG data exist, but their optimal usage in a PAP control algorithm has not been elaborated on. Despite their benefits, normalization of the raw sEMG data and usage of sliding window might be problematic for a real time application. The main aim of this thesis was to develop and test control algorithms for a PAP that successfully distinguishes step-up and step-down tasks of healthy population using exclusively sEMG. A specific aim was to assess the effects of normalization and windowing procedures of sEMG data on prediction success of the algorithms developed. An open database was used to acquire sEMG data of 50 participants. Four machine learning techniques namely, (i) ANN, (ii) LSTM, (iii) LightGBM and (iv) RF were used with windowed/non-windowed and normalized-non-normalized datasets. Comparison of their performances showed that LightGBM utilizing non-normalized and non-windowed dataset performed,(Accuracy=92.53%) not significantly different than the best performance obtained utilizing normalized and windowed dataset(Accuracy=94.02%). So, it is concluded that LightGBM can be used in step-up/down classification module of the control algorithm for PAP using non-normalized sEMG data.

Keywords: EMG Signal, Machine Learning, Decision Trees, Deep Neural Networks

ÖZET

YÜZEY ELEKTROMİYOGRAFİSİ İLE ÖZEL OLARAK BESLENEREK VE YAPAY ÖĞRENME YÖNTEMLERİ KULLANARAK YUKARI-ADIM VE AŞAĞI-ADIM HAREKETLERİ İÇİN AKTİF BİLEK PROTEZİ SINIFLANDIRMA ALGORİTMALARININ GELİŞTİRİLMEŞİ

Yapay zeka alanında olan gelişmeler, hareketi destekleyici cihazlar için sonuçlar doğurmaktadır. Güçlendirilmişş ayak bileği protezleri (GABP), ampute hareketlerini desteklemeyi amaçlayan mekanizmaları kullanır. Yüzey elektromiyografisi (sEMG), algoritmaya kasların katkısına izin verir ve sensör kaynağı olarak kullanımları yenidir. Ham sEMG verilerinin işlenmesi için farklı teknikler mevcuttur, ancak bunların bir GABP kontrol algoritmasında optimal kullanımları üzerinde ayrıntılı olarak durulmamıştır. Yararlarına rağmen, sEMG verilerini normalleştirme ve kayan pencere gerçek uygulamada sorunlu olabilir. Bu tezin temel amacı, yalnızca sEMG kullanarak sağlıklı popülasyonun yukarı-adım ve aşağı-adım görevlerini başarıyla ayırt eden bir GABP için kontrol algoritmaları geliştirmek ve test etmektir. Spesifik bir amaç, sEMG verilerinin normalleştirme ve pencereleme prosedürlerinin, geliştirilen algoritmaların tahmin başarısı üzerindeki etkilerini değerlendirmekti. 50 katılımcının sEMG verilerini içeren bir veritabanı kullanıldı. (i) ANN, (ii) LSTM, (iii) LightGBM ve (iv) RF olmak üzere 4 yapay öğrenme tekniği kullanılmıştır. Performans karşılaştırılması, normalize edilmemiş ve penceresiz veri kümesi kullanan LightGBM'nin, normalleştirilmiş ve pencereli veri kümesi için olan en iyi performanstan (Doğruluk=%94.02) önemli farkı olmadığını gösterdi (Doğruluk=%92.53). Bu nedenle, GABP için normalleştirilmemiş sEMG verileri kullanılarak geliştirilen kontrol algoritmasının yukarı/aşağı adım sınıflandırma modülünde LightGBM kullanılabileceği sonucuna varılmıştır.

Anahtar Sözcükler: EMG Sinyali, Yapay öğrenme, Karar Ağaçları,

TABLE OF CONTENTS

ACKNOWLEDGMENTS . . . iii
ACADEMIC ETHICS AND INTEGRITY STATEMENT . . . iv
ABSTRACT . . . v
ÖZET . . . vi
LIST OF FIGURES . . . ix
LIST OF TABLES . . . x
LIST OF SYMBOLS . . . xii
LIST OF ABBREVIATIONS . . . xiii
1. INTRODUCTION . . . 1
1.1 Muscular System Of Human Lower Limbs . . . 1
1.2 Powered Ankle Prostheses . . . 3
1.3 Identification of movement types . . . 5
1.4 Surface Electromyography . . . 7
1.5 Machine Learning Approach . . . 9
1.5.1 Time-Frequency Analysis . . . 9
1.5.2 Data Preprocessing . . . 11
1.5.3 Feature Extraction . . . 13
1.5.4 Classification algorithms . . . 15
1.5.4.1 Neural Network-based algorithms . . . 16
1.5.4.2 Tree-based algorithms . . . 18
1.6 Gaps in the state-of-the-art . . . 21
1.7 Aims . . . 22
2. METHODS . . . 23
2.1 Dataset . . . 23
2.2 Algorithm development pipeline . . . 26
2.2.1 Preprocessing of sEMG data . . . 26
2.2.2 Feature Extraction of sEMG data . . . 28
2.2.3 Feature Selection of sEMG data . . . 32
2.3 Modelling of sEMG data . . . 34

2.4 Postprocessing 35
2.4.1 Hyperparameter Selection 35
2.4.2 Performance Evaluation 35
3. RESULTS 37
3.1 LightGBM Results 37
3.2 LSTM Results 38
3.3 RF Results 40
3.4 ANN Results 42
3.5 Best performing algorithm selection 43
4. DISCUSSION 45
4.1 Comparison of the classification algorithms 45
4.2 Future Work 48
5. CONCLUSION 52
REFERENCES 53

LIST OF FIGURES

Figure 1.1	Anterior and posterior views of the major muscles on the lower limb.	2
Figure 1.2	A powered ankle prosthesis.	4
Figure 1.3	Step-Up and Step Down kinematic requirements.	6
Figure 1.4	sEMG data collection procedure visualization.	8
Figure 1.5	Transformation of a signal from time domain to frequency domain with FFT.	11
Figure 1.6	Implementation of the different filter types applied to the same data with a same order.	12
Figure 1.7	A neural network architecture.	17
Figure 1.8	Loop of an RNN.	18
Figure 1.9	LSTM structure.	18
Figure 1.10	A decision tree example.	19
Figure 1.11	A random forest classifier.	20
Figure 1.12	Level-wise and leaf-wise tree growth.	21
Figure 2.1	Distribution of each task among 1615 trials.	24
Figure 2.2	Distribution of age groups and gender among 1615 trials.	24
Figure 2.3	Sampling Frequencies' distribution of sEMG data among 1615 trials.	25
Figure 2.4	Flowchart of the feature engineering and modelling process.	26
Figure 2.5	Comparison of raw and filtered sEMG Data of Subject 5.	27
Figure 2.6	Selected Fast Fourier Analyses of sEMG data of Subjects 1, 2, 5 and 7.	28
Figure 3.1	ROC curve of the classification results of all models with the best performing feature extraction selections.	44

LIST OF TABLES

Table 3.1	Confusion table of LightGBM model for Windowed and Normalized dataset.	37
Table 3.2	Confusion table of LightGBM model for Non-Windowed and Normalized dataset.	37
Table 3.3	Confusion table of LightGBM model for Windowed and Non-Normalized dataset.	38
Table 3.4	Confusion table of LightGBM model for Non-Windowed and Non-Normalized dataset.	38
Table 3.5	Confusion table of LSTM model for Windowed and Normalized dataset.	39
Table 3.6	Confusion table of LSTM model for Non-Windowed and Normalized dataset.	39
Table 3.7	Confusion table of LSTM model for Windowed and Non-Normalized dataset.	39
Table 3.8	Confusion table of LSTM model for Non-Windowed and Non-Normalized dataset.	40
Table 3.9	Confusion table of Random Forest model for Windowed and Normalized dataset.	40
Table 3.10	Confusion table of Random Forest model for Non-Windowed and Normalized dataset.	41
Table 3.11	Confusion table of Random Forest model for Windowed and Non-Normalized dataset.	41
Table 3.12	Confusion table of Random Forest model for Non-Windowed and Non-Normalized dataset.	41
Table 3.13	Confusion table of ANN model for Windowed and Normalized dataset.	42
Table 3.14	Confusion table of ANN model for Non-Windowed and Normalized dataset.	42

Table 3.15 Confusion table of ANN model for Windowed and Non-Normalized dataset. 42
Table 3.16 Confusion table of ANN model for Non-Windowed and Non-Normalized dataset. 43
Table 3.17 Accuracy percentages of four models with respect to changing feature extraction methods. 43

LIST OF SYMBOLS

a_0	Scaling factor of Wavelet transform
b_0	Time shift factor of Wavelet transform
ξ	Frequency in FFT
n	The degree of the Butterworth filter
w_c	Cut-off frequency of low-pass Butterworth filter
G_0	Gain of Butterworth filter in zero frequency
ϵ	Threshold
σ	Activation function of ANN
Vi_j	Weights of ANN
x_j	Inputs of ANN
Vi_j	Weights of ANN
T_i^{hid}	Bias terms of hidden units of ANN

LIST OF ABBREVIATIONS

EMG	Electromyography
sEMG	Surface Electromyography
ML	Machine Learning
AI	Artificial Intelligence
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
DNN	Deep Neural Network
LSTM	Long Short-Term Memory
LightGBM	Light Gradient Boosting Machine
RNN	Recurrent Neural Network
RF	Random Forest
SVM	Support Vector Machine
XGBoost	Extreme Gradient Boosting
FFT	Fast Fourier Transform
TA	Tibialis Anterior
GM	Gastrocnemius Medialis
SO	Soleus
PL	Peroneus Longus
VM	Vastus Medialis
BF	Biceps Femoris
GMax	Gluteus Maximus
RF	Rectus Femoris
WAM	Willison Amplitude
RMS	Root Mean Squared
MAV	Mean Absolute Value
ZCR	Zero Crossing
LOD	Log Detector

1. INTRODUCTION

1.1 Muscular System Of Human Lower Limbs

Human musculoskeletal system has a very complex architecture. For many years, researchers have tried to have a better understanding of it. The anatomy of the lower limb of a human is also a complex system and a better understanding of the anatomy of the human lower limb may lead to developing prosthetic devices for the lower limb closer to the similar functioning of the amputated organ. Figure 1.1 shows the anterior and posterior views of the major muscles on the lower limb [1].

There had been different approaches to the functioning of the muscle unit. Hill's muscle model showed that there is both active and passive force generation in the muscles. There are different models about the lower limb's musculoskeletal system utilizing Hill's model [2], [3]. As there is more knowledge about the anatomy of the musculoskeletal system, different models will enable different prosthetic investigations.

Different types of muscles and tendons which have complex interactions with each other exist in both the lower and upper parts of the lower limb of the human. The largest tendon in the body is the Achilles tendon which connects heel bone and calf muscles by stretching. Tibialis Anterior muscle is placed on the lateral side of the tibia bone and it is the main dorsiflexor muscle of the foot. It has a critical function in lifting the foot by dorsiflexion during locomotion. Gastrocnemius Medialis muscle is placed behind the femur in the anterior view with two heads from the top and forms Achilles tendon by fusing with Soleus muscle. Soleus muscle creates plantar flexion and is placed in the posterior leg. Peroneus Longus creates plantarflexion and is placed in the lateral part of the leg. Tibialis Anterior, Gastrocnemius Medialis, Soleus, and Peroneus Longus muscles are placed in the lower part of the lower limb. Vastus Medialis is a knee extensor muscle which is one of four muscles forming quadriceps femoris muscle, it is placed medially in the thigh. Rectus Femoris is also one of four muscles

forming quadriceps femoris muscle and is placed in anterior of the thigh, it is the flexor of the thigh. Biceps Femoris is placed in the posterior of the thigh and it is the flexor of the knee and creates an external rotation. Gluteus Maximus is placed in the posterior aspect of the hip joint and it creates an extensor of the thigh. Vastus Medialis, Rectus Femoris, Biceps Femoris, and Gluteus Maximus muscles are placed in the upper part of the lower limb [4], [5], [6]. These muscles and the other muscle and tendon groups are interconnected and help humans to perform locomotion satisfactorily.

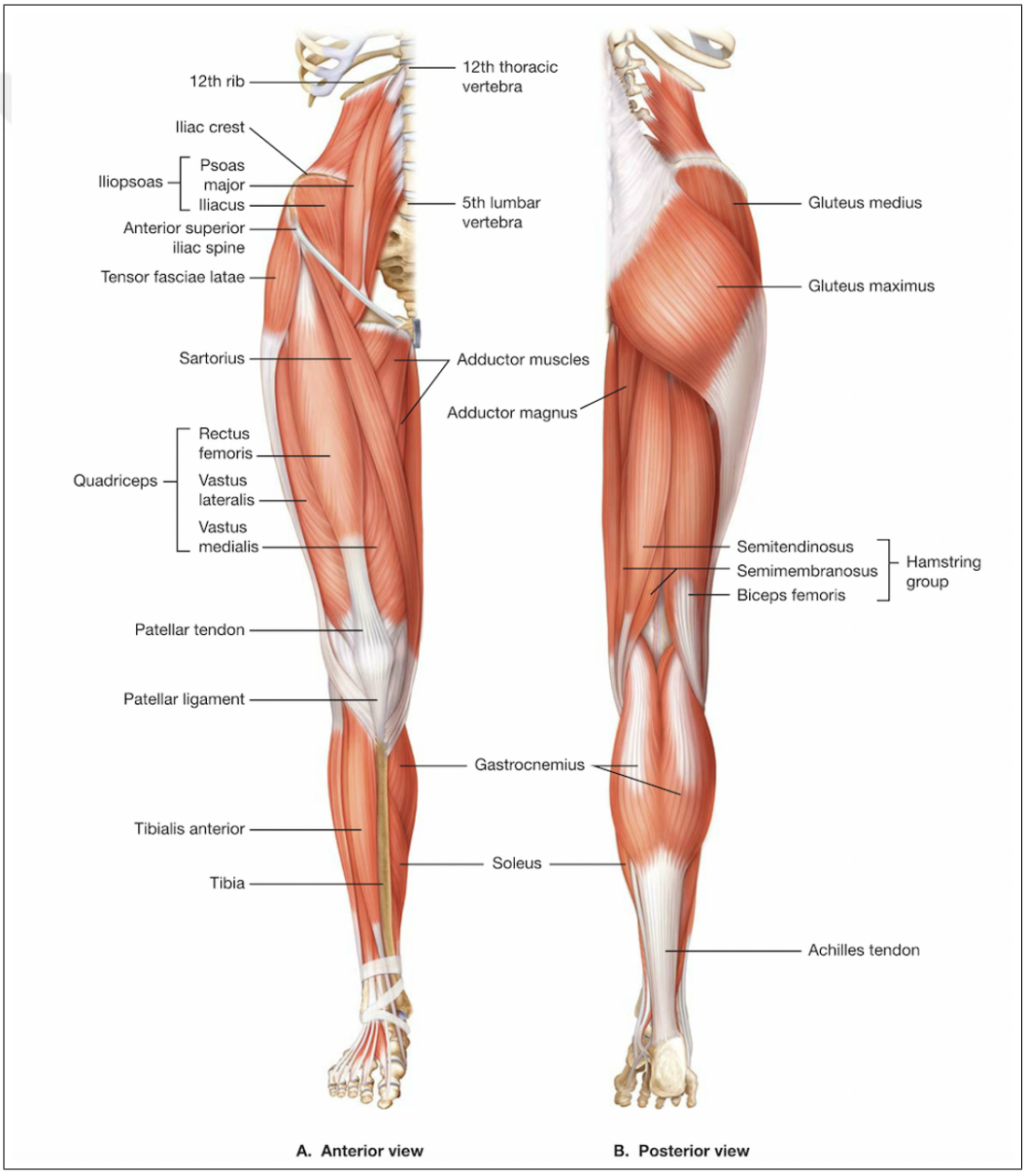


Figure 1.1 Anterior and posterior views of the major muscles on the lower limb adopted from [1].

1.2 Powered Ankle Prostheses

Prostheses are human-made devices that are used to replicate the anatomical behavior of the human biomechanics. There are different types of prostheses that replace different parts of the human body like the arm, foot, and even face. Also, there are different types of prostheses which are passive, variable damping, and powered. In this thesis, the main focus will be on the workings of powered prostheses for the lower limb.

The powered ankle prostheses, which are also known by the name active ankle prostheses, are devices inserted in the amputated ankle and use actuation power to maintain the needed forces to assist amputees while locomotion. The actuation mechanism is the most important part of the powered limb prostheses because it makes the main difference. There are different design options to be used in the actuation of the powered ankle prostheses. Some of these actuation options are Electromechanical, Pneumatic, Hydraulic, and Variable impedance actuators [7], also some prostheses use hybrid actuation systems that combine different features of the different actuators [8] and have very successful results. Each of these actuation mechanisms has a different working principle and different advantages and disadvantages over themselves.

The prostheses technology has become more complex as the years have passed and with the development of artificial intelligence, developments have also started to be applied for prostheses. Improvement in understanding the relationship between muscles and the movement types may improve the way designing more efficient and more reliable prostheses [9]. Powered prostheses are devices that can use this understanding to successfully implementing the necessary torques to its actuators to accomplish the intended task satisfactorily [10]. Through developments in artificial intelligence, position and the moment can be estimated accurately in some studies [11] which is advantageous in designing powered ankle prostheses. More developed powered ankle prostheses can be achieved through improved knowledge about musculoskeletal morphology and designing actuators with the help of the control system more economically [12].

There is a need to have a wider understanding of gait patterns to get more advanced powered ankle prostheses. Figure 1.2 shows an implementation of a designed powered ankle prosthesis of Grimmer et al. [13].



Figure 1.2 A powered ankle prosthesis adopted from [13].

Deep learning is a newly emerged area and its applications in many areas have shown its success in processing non-tabular data like image and sound processing. Time series data like joint kinematics during locomotion were seen to be processed successfully by several deep learning methods [14] and its applicability is not limited only to processing kinematic data.

In controlling the powered ankle prostheses, deep learning methods can provide an advantage in detecting gait patterns, movement types, and movement intentions. In theory, deep learning methods' advantages are sourced from their ability to directly processing the input data without requiring any preprocessing. There are some studies that use this advantage to classify gait patterns and movement types with the data collected from both IMU sensors and EMG signals [15], [16]. The studies implying deep learning methods onto powered prostheses will show its impact on gait-related classifications with more applications as the years passed.

1.3 Identification of movement types

Powered ankle prostheses are robotic devices that help amputees in their daily lives by mimicking the dynamics of the musculoskeletal system of humans. The forces and the moments required to perform locomotion are produced by powered prostheses during gait by means of active actuation mechanisms [3]. There are numerous movement types including but not limited to walking, running, stair ascending-descending, step-backing, jumping. Even though we do not realize in our daily lives, each of these movement types requires different muscle groups' activation to successfully accomplish the desired movement [6]. Thus, it is important to have a better understanding of muscles' hierarchy while performing specific movements.

Figure 1.3 shows the different kinematic requirements in step-up and step-down tasks. There are different needs in the swing and stance phases performing these movements. While performing Step Up task, SO, GMax, VM, RF, and Hamstring muscle contract to raise up the body. GMax also prevents the falling of the body. TA muscle supports the limb during the swing phase and Hamstrings control the knee by flexing at the end of the movement. While performing Step-Down, SO, VM, RF lengthen to lower on the body. Again, GMax prevents the falling of the body. During other phases and parts of the phases, the different duties of the muscles dissociate [17].

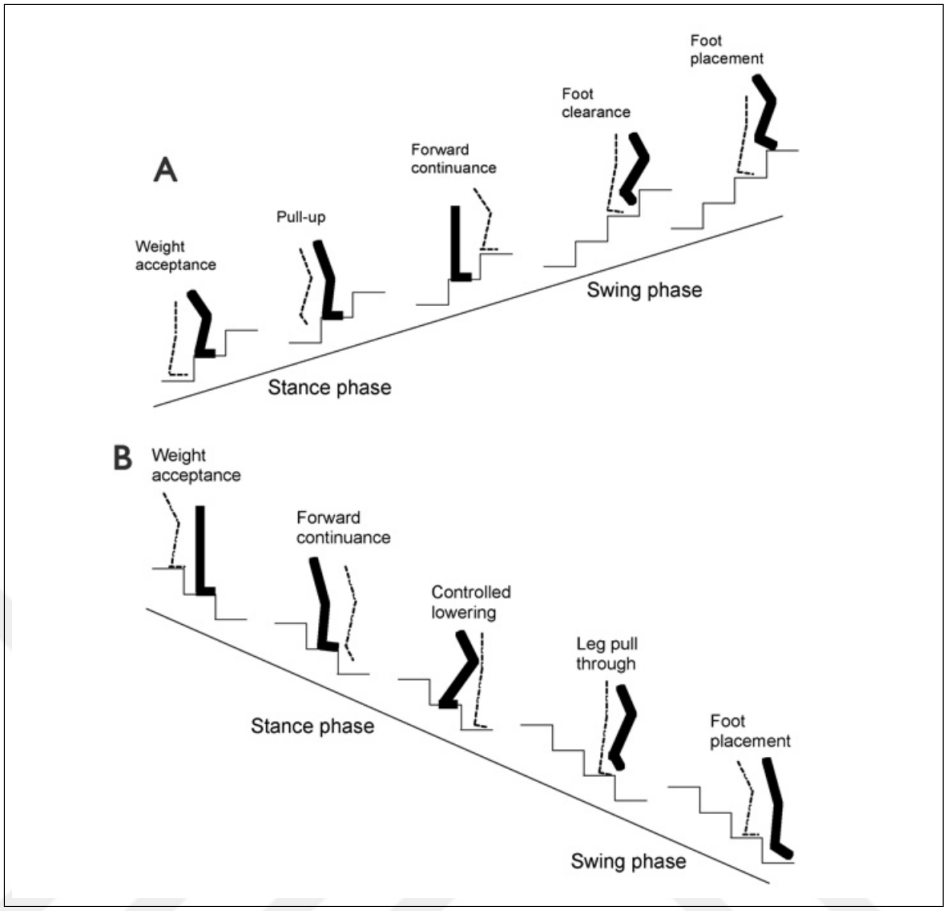


Figure 1.3 Step-Up and Step Down kinematic requirements [17].

One way of understanding the relationship between muscles and the movement types can be through dissociating the gait characteristics of movement types. The gait patterns in human locomotion have been investigated in many aspects [18]. In some studies, the gait patterns are investigated through kinematic analyses of the joints in the human body [19], [20].

On the other hand, some studies also investigated the EMG Signal characteristics of the muscles in the lower limb [21], [22]. Also, there are some studies that use sensor fusion methodology to analyze gait patterns by combining EMG Signals, cameras and IMU's and Pressure sensors [23], [24], [25] to get a better understanding of the gait patterns.

Classification of the movement types for powered ankle prostheses is very crucial for the comfort and safety of amputees. Each of the movement types has different dynamical requirements and requires different muscles to perform the intended movement accurately. By classifying the movement types from each other, the control algorithms may have an attention mechanism and work more properly. There are many studies in the literature both for specific movement types like Step-Up&Down [26], [27], Walking, Running [28],and for recognizing movement intentions entirely [14], [29].

1.4 Surface Electromyography

Electromyography (EMG) is a diagnostic method to measure electrical activity and response of the muscle cells during stimulation. The activity of the muscles can be seen in the measurements before 80-100 ms before the contraction occurs [30]. The powered prostheses of the amputees can be controlled by means of using the EMG signal as a measurement method. Muscles are contracted with the electrical signals coming from the brain through peripheral motor nerves. The muscles are contracted only if the signals are targeting them [31]. EMG signal measurement provides information about the intended movement or targeted muscles before the targeted muscles are contracted or the intended movement is made. Thus, the utilization of EMG signal for controlling powered prostheses is advantageous as it gives information just before the intended movement.

Surface Electromyography(sEMG) is a non-invasive procedure and similar to EMG collects the electrical activity coming to the muscles. It collects the muscle activity data through the skin and has different advantages and disadvantages over invasive EMG collection procedures. Skin conditions may have an adverse effect on the quality of data on the surface-EMG collection, however, invasive EMG collection procedure may have a negative effect on comfort and has different disadvantages during real-time usage on amputees.

Figure 1.4 shows the sEMG data collection procedure. The electrical activity coming from the spinal cord through motor neurons is captured via bi-polar electrode contacts and the electrical activities are converted into a signal form. sEMG data in the time domain are gathered as the voltages correspond to the magnitude of the electrical signals.

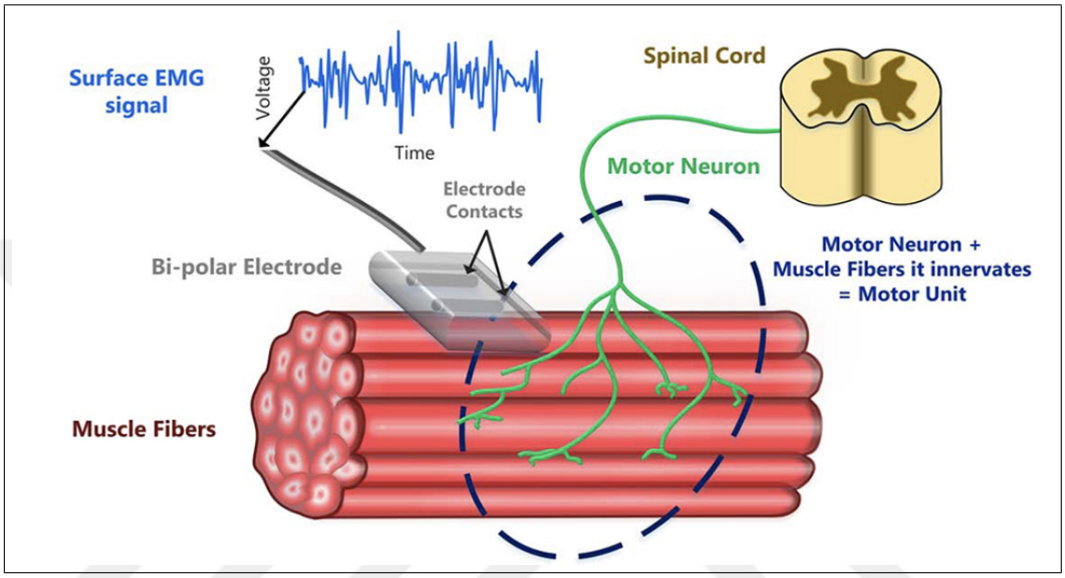


Figure 1.4 sEMG data collection procedure visualization [31].

sEMG data are widely used in processing muscle signals. There are a lot of studies in which sEMG data are used for hand gesture classification and controlling prosthetic hand [32]. sEMG data are also widely used for lower limb in detecting patterns of gait cycles and disorders, classification of the movement types and controlling the powered prostheses [33], [34]. In recent years, different processing algorithms and controlling schemas advanced the sEMG data usage in controlling powered lower limb.

For most cases, sEMG data are recorded with non-invasive methods and it is also advantageous in applications of powered prostheses. Analyzing the sEMG data can be very complicated due to its complex nature. Additionally, there are plenty of noises sourced from external factors while gathering the sEMG data.

These factors can arise from both physiological reasons like tearing, vibrating inappropriate skin conditions, or external factors like the weather and electromagnetic equipment problems [35]. Thus, before feeding sEMG data into the processing modules, the data should be preprocessed to reduce the noise so as not to let these noises ruin the model stability.

1.5 Machine Learning Approach

1.5.1 Time-Frequency Analysis

In recent years, the developments in Artificial Intelligence (AI) and in signal processing techniques made more accurate sEMG processing and detection studies possible. To analyze a signal like sEMG, time-frequency analyses should be used because the time and frequency domains of the signals are interrelated but different from each other. To process a signal in a time domain, analyses in the frequency domain should be performed so as to reveal important statistics which change in the time domain. There are different types of time-frequency analysis that can be implied for both stationary and non-stationary signals [35]. These analyses can also be done for understanding the functional frequency range of the recorded signal. This makes it possible to know the frequency bands in which the noises are prominent and the bands in which the important time-domain statistics are present. With the help of the time-frequency analysis, the band-pass frequencies can be obtained more reliably and the filtering operations for reducing the noise can be done accordingly [36].

There are different types of time-frequency analysis that use different approaches and formulations than each other. Wigner-Ville Distribution, Hilbert Huang transform, Power spectrum density functions, Continuous and Discrete Wavelet Transforms, Short and Fast Fourier transforms are some of these analyses which use different formulations and have different computational complexities [37], [38].

Even though Fourier transform methods have extensively been used in time-frequency analyses, in recent years Wavelet analyses have also been popular because the Fourier transforms assume the signal as a stationary signal [39]. For sEMG data, there is not a common choice between them and both Wavelet analyses and Fourier transforms are preferred to analyze the time-frequency domain.

The Discrete Wavelet Transform allows a signal to preserve the quality and properties, acting in a numerically stable way causing less information redundancy [40]. Its realization takes place through successive digital filtering on the original signal, where the digital filters are defined as filters which are described by the functions h(n) and g(n), which are the impulse response functions of a low-pass and high-pass filter respectively, each with half the band of the original signal. It is mathematically defined as $X_{m,n}(a,b)$ in Eq. 1.1 where a_0 is the scaling factor and b_0 is the time shift factor.

$$X_{m,n}(a,b) = \frac{1}{\sqrt{a_0^i}} \int_{-\infty}^{\infty} x(t)\Psi(a_0^{-i}t - nb_0) \tag{1.1}$$

The Fast Fourier Transform (FFT) converts the sEMG data from the time domain to the frequency domain to see the critical frequency ranges [36]. FFT computes the inverse of discrete transform and FFT is widely used in signal processing of sEMG. Digital filters created to eliminate the noise use frequency bands to do filtering. By doing FFT analysis, the important frequency ranges can be found. The mathematical formulation is stated in Eq. 1.2 and ξ represents the frequency [41]. Figure 1.5 shows an example of time-frequency transformation with FFT.

$$\hat{y}(\xi) = \int_{-\infty}^{\infty} y(x)e^{-2\pi i x\xi}\, dx \tag{1.2}$$

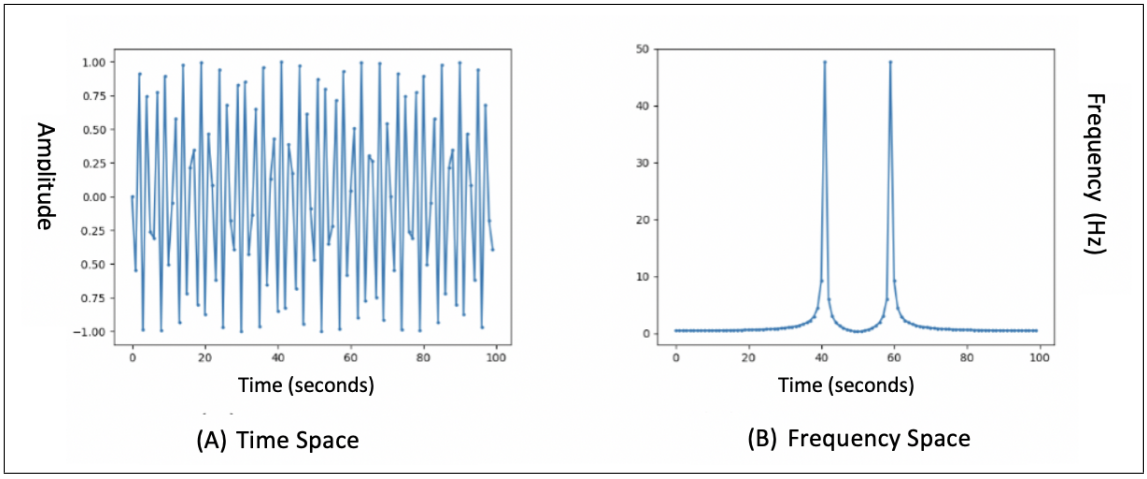


Figure 1.5 Transformation of a signal from time domain to frequency domain with FFT.

1.5.2 Data Preprocessing

There are different data preprocessing techniques that are applied to the raw form of the sEMG data. Filtering the raw data is the first step in the preprocessing stage in order to remove the noise. Analog filters and digital filters are the two types of filters applied to the signals. There are different types of filter designs for digital filters used in sEMG processing studies. Chebyshev, Bessel, Eliptic, and Butterworth filters use different formulations and are extensively used in sEMG data filtering [42]. Figure 1.6 shows the implementation of the different filter types applied to the same data with the same order [43].

Butterworth filter is the most used filter design in sEMG data filtering to eliminate the noise. It is a filter that smoothes the frequency response of a signal in the desired frequency ranges [44]. Eq. 1.3 shows the formulation of a low-pass butterworth filter where n is the degree of the butterworth filter, w_c is the cut-off frequency and G_0 is the gain in the zero frequency.

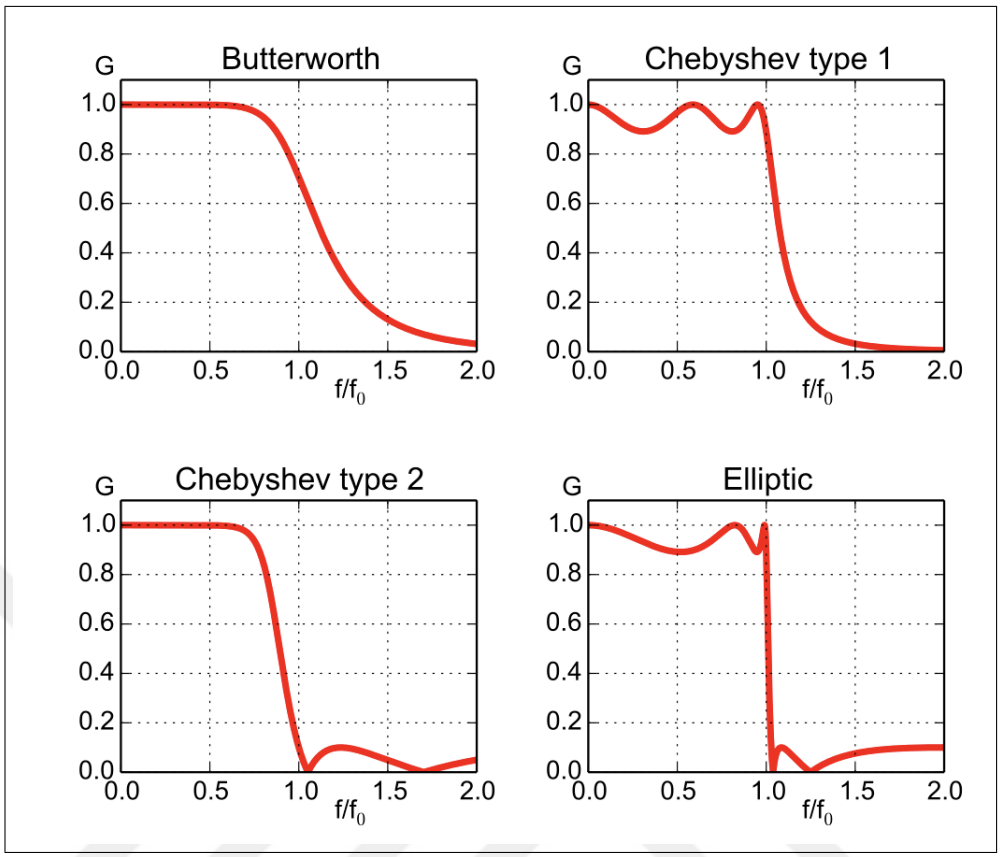


Figure 1.6 Implementation of the different filter types applied to the same data with a same order [43].

In the band-pass version of this filter, a low-pass butterworth filter is combined with a high-pass butterworth filter. In this case, the frequency ranges between low and high pass cut-off frequencies are filtered.

$$G(w) = \frac{G_0}{\sqrt{1 + (\frac{jw}{jw_c})^{2n}}} \tag{1.3}$$

Another preprocessing technique that is extensively used for sEMG data is normalization. Normalization is a widely used preprocessing technique that aims to bring the data values (individually or entirely) into the same range. Normalization of sEMG data is needed for some Machine Learning (ML) algorithms for having successful results.

There are many successful results in classification of sEMG data in the literature [45], [46]. On the other hand, there are also some studies for the classification of sEMG data that recommends normalization does not have to be used [47]. Normalization is not necessary for all of the ML algorithms, but some of the algorithms' way of working requires normalizing the data.

One type of ML algorithm needing normalization is neural network-based algorithms like Artificial Neural Network (ANN), Recurrent Neural Network(RNN), Long Short-Term Memory (LSTM), etc. Besides generally speeding up the learning and leading to faster convergence, the vital reason for using normalization is that gradients are getting larger and larger for non-normalized and numerically large inputs while backpropagation damages the learning process. However, in some cases, the neural networks may not be damaged from not normalizing, depending on the input values' range and the learning task.

One type of ML algorithm which does not need normalization is decision tree-based algorithms like Random Forest (RF), Extreme Gradient Boosting(XGBoost), Light Gradient Boosting Machine (LightGBM), etc. In the decision tree and in the other tree-based ensemble methods normalization is not required, because they take decisions on the basis of the individual parameters while branching down the tree and each branch is independent of the other, so variation in the inputs has no impact on the performance of these models.

1.5.3 Feature Extraction

sEMG data have a very complex structure because the electrical signal coming through motor nerves can be combinations of the signals going to different muscles. Requirement of choosing appropriate action for the each received sEMG data come with a need of the model to understand the signal meanings correctly.

To understand the complex behavior of the sEMG data, feature extraction techniques are used for having more information from the same sEMG data. There are several options for extracting features from the raw form of the sEMG data [48]. The actionable information from the signal can be gathered by extracting meaningful features from the raw data.

Sliding window theorem is one of the widely used methods while extracting features from time-series data like sEMG. In the sliding window theorem, an imaginary window with the desired window size, which corresponds to a timestamp, is created and is mapped onto the real data from the beginning to the end sequentially [49]. There is a wide range of algorithms that are applied to each window some of them are Simple Square Integral (Sş), Root Mean Squared (RMS), Zero Crossing (ZCR), and Mean Absolute Error (MAE). There are hundreds of different feature extraction algorithms which are used in the literature [50], [51]. These algorithms are used to extract useful information from the raw or the preprocessed data, also each of these algorithms has different mathematical expressions. Thus, different mapping properties of these algorithms lead to obtaining various numerical results from the same data.

For deep learning algorithms, feature extraction is not required for having successful results. Even though extracting features helps models to be more interpretive, they can cope with complex data and the performances of the models are not reduced significantly due to the nature of the deep learning algorithms as they extract meaningful information on themselves [46]. Sequence models like RNN and LSTM have specialized designs to cope with sequential data like sEMG data and even though they use raw sEMG data they have similar performances with the models which use feature extraction [52].

1.5.4 Classification algorithms

There are many Signal Classification algorithms used for identifying and classifying sEMG data. Most of these algorithms performed with high accuracies. The traditional machine learning algorithms' performances are impactful. J. Liu and Zhou et al [53] studied two machine learning algorithms, Linear Discriminant Analysis and KNN, for developing pattern recognition strategy for hand functioning, and their model performed with 97% accuracy. Stango et al [54] used a support vector machine (SVM) to classify patterns in the upper limb and the classifier performed with 95% accuracy. Tavakoli et al [55] also used an SVM classifier for classifying hand gestures and they performed 90% accuracy overall. Diana C. et al [27] classified movements on the lower limb based on the sEMG data with SVM and that study's accuracy is more than 90% Kuang-Yow et al [56] used a multimodel approach with decision tree and KNN for hand gesture recognition and their model performed with 89% accuracy.

Deep learning algorithms have also been used extensively since their applications in different areas became state of the art. There are many studies about sEMG classification for both the lower and the upper limbs. Morbidoni et al [21] used Multilayer Perceptrons for classifying gait phases during walking and the study performed with 93% accuracy. Xia et al [29] proposed a model which combines Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) to estimate limb movement and the accuracy of that study is 95%. Allard et al [57] proposed a CNN network for classifying hand gestures and they performed with online accuracy of 98%. Lu et al [58] proposed an LSTM algorithm to recognize the jump phase of the human. The study was performed with 97% accuracy. There are more studies show the impact of machine learning algorithms in signal processing and more accurate models will be developed as the new ML algorithms and AI techniques will emerge in the following years.

1.5.4.1 Neural Network-based algorithms. ANN is an algorithm that forms the basis of deep learning, a subfield of machine learning, which was developed with the aim of creating machines that can imitate the human brain. ANNs consist of connected groups of artificial neurons. These neurons acquire, store, and use knowledge with experience.

ANN architecture is comprised of three interconnected layers which are called input, hidden, and output layers. Each connection is represented with numeric numbers called weights. In the hidden layer, the output h_i of neuron i is calculated as in Eq. 1.4

$$h_i = \sigma(\sum_{j=1}^{N} V_{ij} x_j + T_i^{hid}) \tag{1.4}$$

where σ is the transfer (activation) function, Vi_j weights, x_j inputs, N number of inputs and T_i^{hid} threshold terms of hidden nodes. The activation function not only brings nonlinearity to the neural network but also limits the calculated values of the units [59].

In every conventional neural network, there are two key phases in the data processing procedure: learning and application. To train a network, a training database is needed in the first phase. The learning phase's goal is to fine-tune the weights of connections between layers. Following the setup of the learning samples, a sample will be fed into the network and the generated outputs will be compared to the known outputs in an iterative way. If the result and the unknown output are not the same, the weights of the connections will be changed until the difference is as little as possible. The validation dataset is used to the network for the validating phase once the learning process has achieved the required convergence for the networks [60].

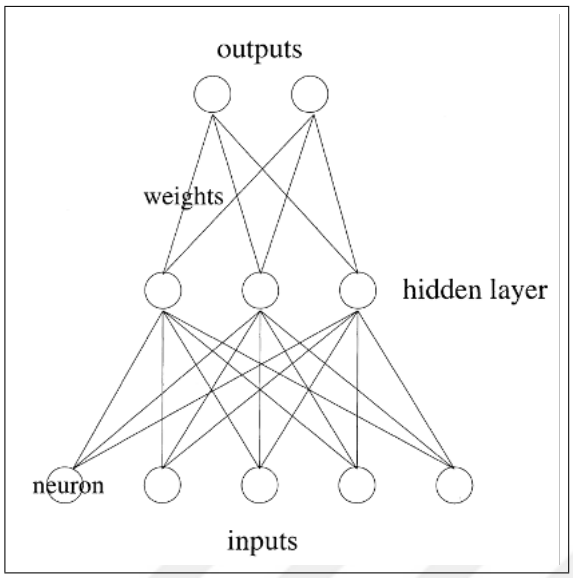


Figure 1.7 A neural network architecture [59].

LSTM networks are a type of recurrent neural networks (RNNs), that is a complex area of deep learning. Recurrent networks can use their feedback connections to store activations as representations of input events, which can be thought of as the short-term memory [61]. Many applications, such as speech processing, non-Markovian control, and music creation benefit from this algorithm [62].

In sequential data, previous knowledge is important. For instance, humans make use of the preceding words in a sentence to understand the current word. Traditional neural networks cannot make use of this sequential knowledge, which is a major disadvantage. However, RNNs issue this problem with the mechanism of loops which lets information be transferred from one step to the next. These loops can be thought of as copies of connected neural networks that pass a message to the next one [63].

LSTMs are a special type of RNN that works much better than traditional RNNs. They can make use of the information from not only the previous step but also the steps before that, giving them the ability to remember information for a long period of time. The structure that enables this capability consists of a cell state and gates.

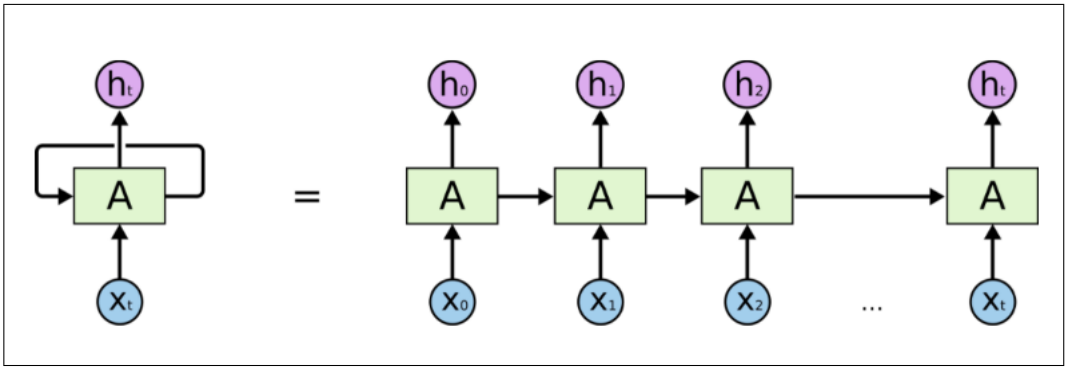


Figure 1.8 Loop of an RNN [63].

A cell state is a conveyer that travels along the chain, and it carries information. LSTM can remove or add information to the cell state, which is regulated by the gate mechanisms. Gates are the decision centrals that let the information get through. They are formed from a sigmoid neural network layer and multiplication operation. A sigmoid layer is a part that decides how much information to let through [63].

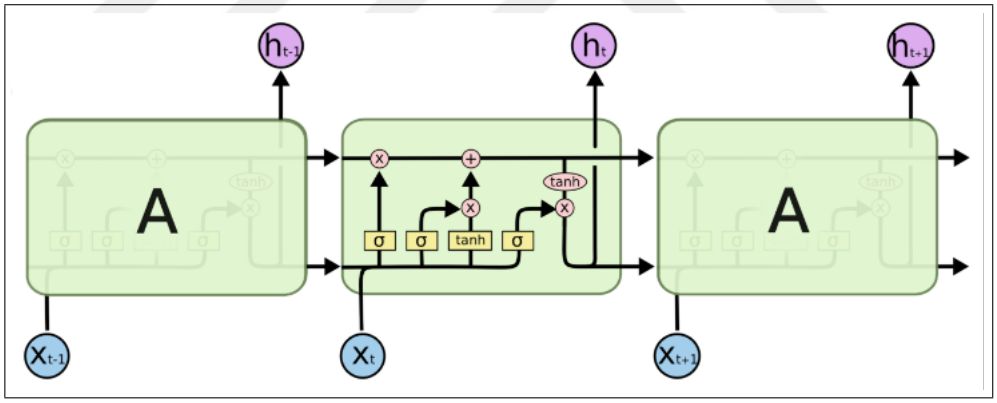


Figure 1.9 LSTM structure [63].

1.5.4.2 Tree-based algorithms. Random Forest is a machine learning algorithm that is made up of many decision trees that consist of questions. Each question is used to arrive at a final decision by acting as decision nodes where the data is split according to the answers. The aim of these trees is to find the best split to subset the data. Ensemble methods that combine multiple decision trees are developed not only to combat the bias and overfitting problems that may arise from using a single tree but also to make more accurate predictions [64].

The most popular ensemble methods are bagging and boosting. In bagging, in other words, bootstrap aggregation, a random sample of training data is selected with replacement. After selecting multiple samples, trees are generated independently, and the average or majority of the predictions are used. Random forest algorithm is an improved version of the bagging method since it uses both bagging and feature randomness, that is, the procedure to generate a random subset of features to ensure low correlation between trees.

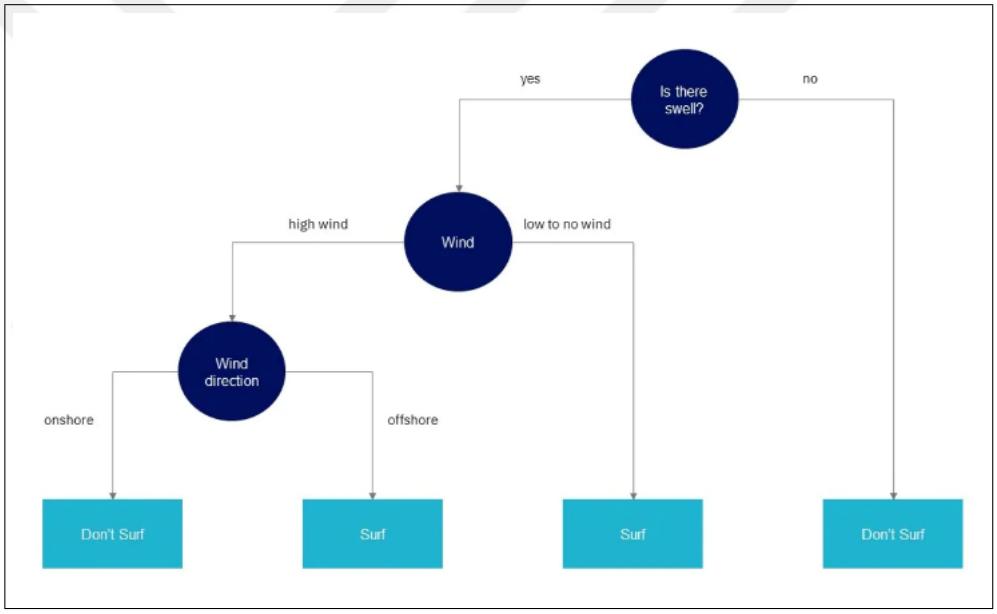


Figure 1.10 A decision tree example [64].

Each tree in a random forest consists of a bootstrap sample, a random data sample that is drawn from the training set. From each sample drawn, one-third of is used as the test set, which is called the out-of-bag sample. After using feature bagging to ensure more diversity to the samples and to reduce the correlation among trees, prediction is made by averaging the results of individual decision trees for regression or by taking the most frequent categorical variable for classification.

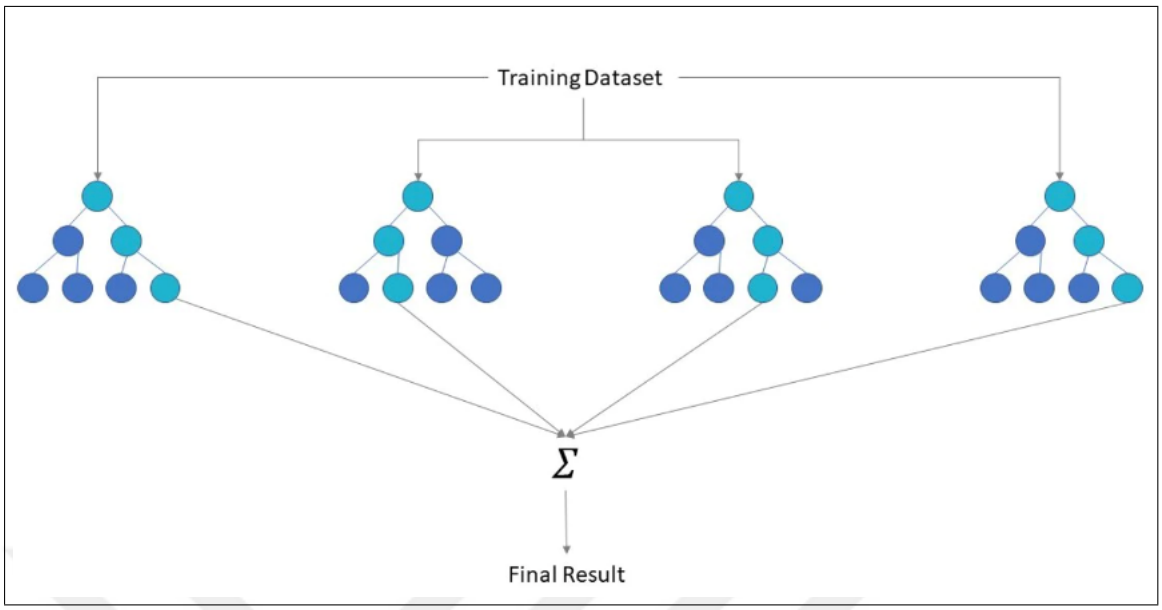


Figure 1.11 A random forest classifier [64].

As stated in the random forest section, there are two main ensemble methods, one of which is boosting, a collection of weak learners that form a strong learner. In boosting, models are trained sequentially using random samples of data where each model tries to compensate for the errors of the preceding model.

There are several boosting algorithms characterized by the process of the creation and aggregation of the weak learners. Gradient boosting is one of the most popular ones which trains using the residual errors of the predecessor model by making use of the gradient descent algorithm [65].

LightGBM is a distributed gradient boosting framework that is faster, more efficient, that use lower memory and result in better accuracy than other gradient boosting algorithms [66]. It is a histogram-based algorithm that puts continuous features into discrete bins. While most methods grow trees depth-wise (level-wise), LightGBM grows them leaf-wise by choosing the leaf with maximum loss and growing the tree based on that leaf, achieving lower loss compared to other methods [67].

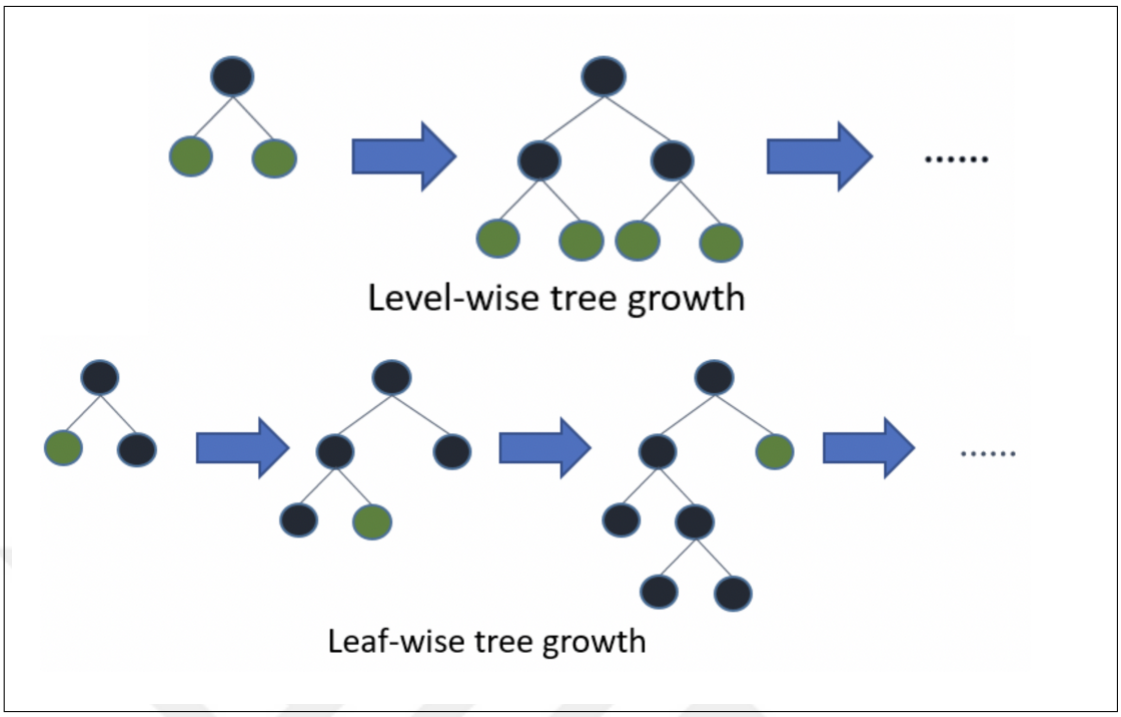


Figure 1.12 Level-wise and leaf-wise tree growth [67].

1.6 Gaps in the state-of-the-art

There is a need to distinguish the movement types for a better prosthesis. There are different studies for movement recognition [14], recognizing movement intention [29] using sEMG data. Also, there studied different control algorithms for different movement types, it is for walking in one study [11]. However, there is no study working for classifying the Step-Up and Step-Down movement types using sEMG data.

The computational complexity and feature extraction procedures affect the cost and the reliability of the powered ankle prostheses for a variety of reasons. There are several feature extraction methods to gather information from the sEMG data and these features can be extracted in either time or frequency domain. While extracting these features, the sliding window method is used in many studies which increases computational complexity. Also, normalization choices affect the models' performance.

Most of the studies which process sEMG data normalize the sEMG data before making a classification. Additionally, the data fed into the model are extracted through feature extraction algorithms which are often applied with windowing procedure. There is no study about the effects of the selections of normalizing and windowing on the classification of movement types.

1.7 Aims

The main aim of the thesis was to develop and test control algorithms for a powered ankle prosthesis that successfully distinguishes Step-Up and Step-Down tasks of healthy population using exclusively sEMG.

In order to achieve the main aim, a specific aim was to assess the effects of normalization and windowing procedures of sEMG data on the success and computational cost of Step-Up and Step-Down classification algorithms developed.

2. METHODS

2.1 Dataset

In order to develop the targeted control algorithms in this thesis, a dataset gathered from healthy subjects was used. Testing these algorithms on amputees usage will be conducted by gathering amputees data afterward. Data were acquired in the Laboratory for Movement Analysis of the Biomedical Technology Department of Don Carlo Gnocchi Foundation Scientific Institute of Milano, Italy. The laboratory is equipped with a 9-camera motion capture system (SMART system, BTS, Garbagnate Milanese, Italy), two force platforms (Kistler, Winterthur, Switzerland), and an 8-channels wireless sEMG recording system (ZeroWirePlus, Cometa, Bareggio, Italy). The dataset used was published by Lencioni et al. [68]

Kinematic, kinetic, and surface electromyography (sEMG) data were recorded in that study for different tasks including level walking, heel walking, toe walking as well as step up and step down motions from fifty healthy subjects. Each of these subjects performed several trials, so from fifty subjects 1615 different trials were recorded. Figure 2.1 shows the distribution of these trials among each locomotor task.

Each of the tasks requires to use of a different group of muscles. In the dataset, sEMG data were recorded from eight different muscles from the lower limb. Four out of these eight muscles were lower leg muscles namely the Tibialis anterior (TA), peroneus longus (PL), gastrocnemius medialis (GM), and soleus (SO). The remaining four out of these eight muscles were upper leg muscles namely the vastus medialis (VM), biceps femoris (BF), gluteus maximus (GMax), and rectus femoris (RF).

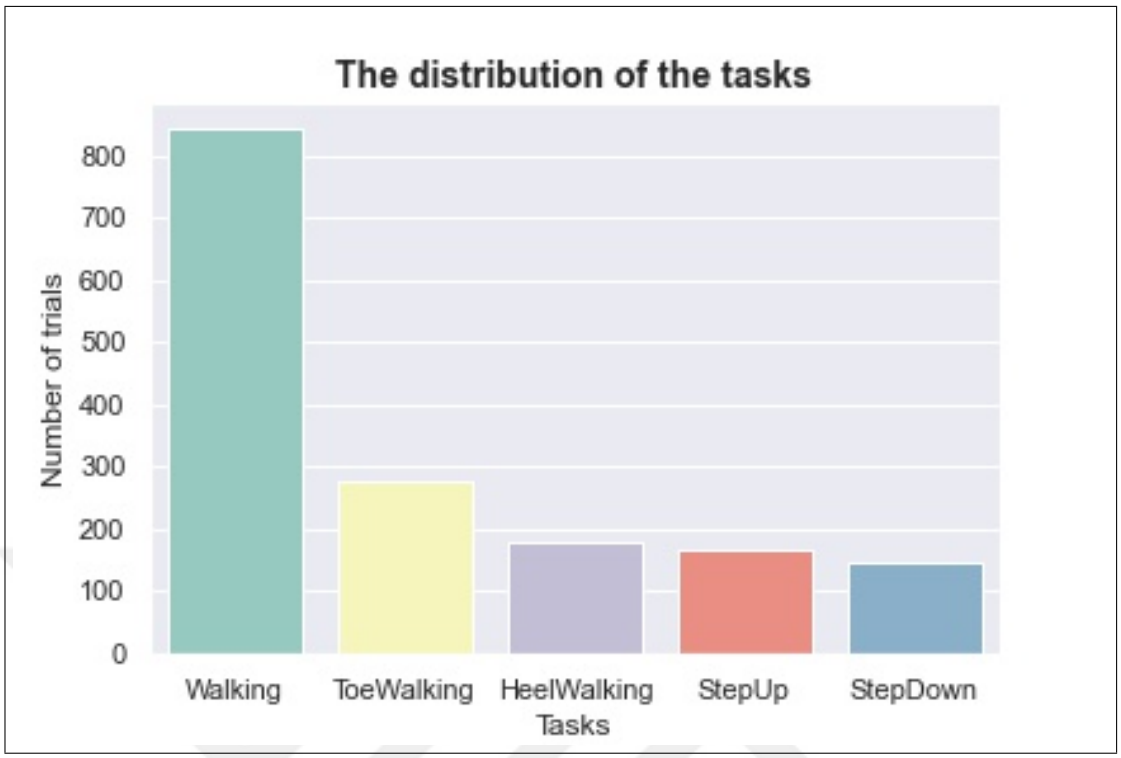


Figure 2.1 Distribution of each task among 1615 trials.

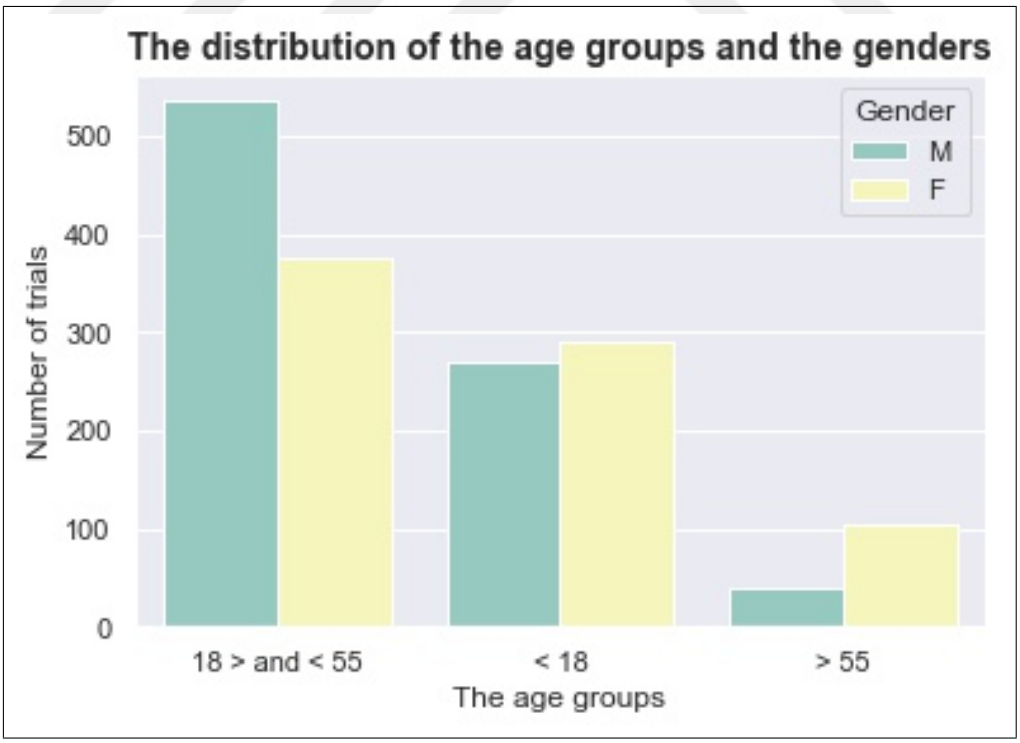


Figure 2.2 Distribution of age groups and gender among 1615 trials.

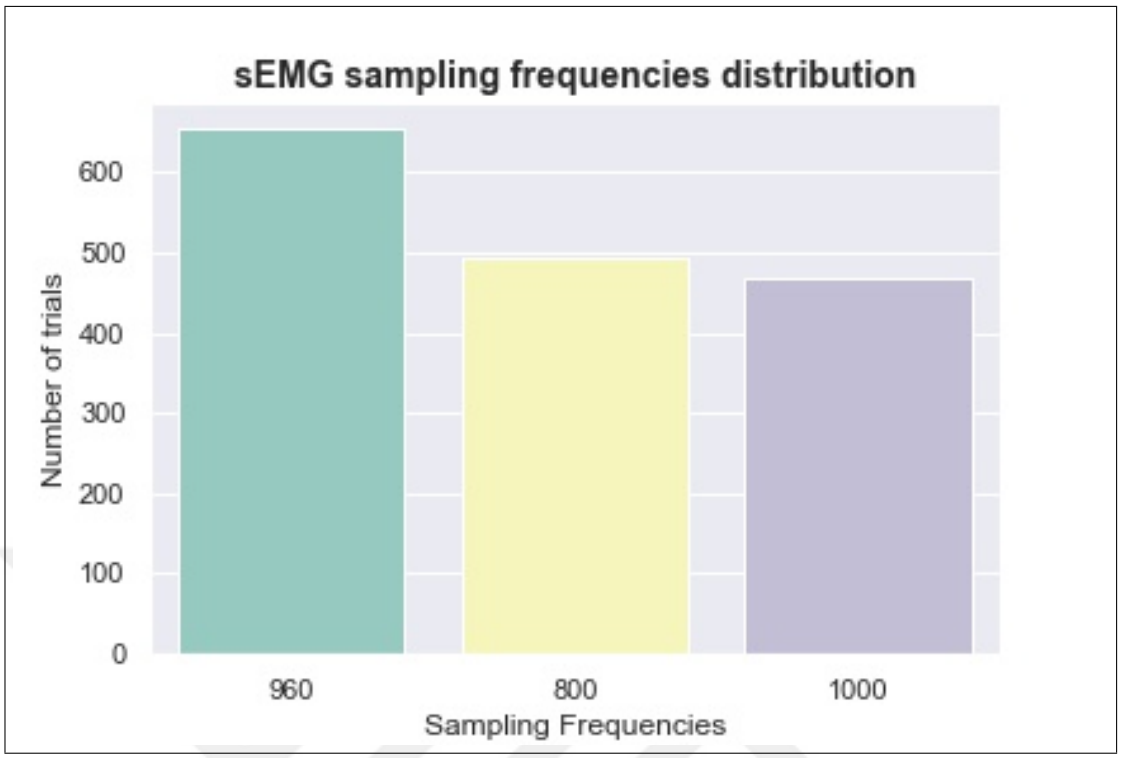


Figure 2.3 Sampling Frequencies' distribution of sEMG data among 1615 trials.

The 50 subjects have differing anthropometrics. Their ages, masses, heights, stride lengths, and genders differ from one person to another. Figure 2.2 demonstrates the distribution of age groups and gender in combination that most of the trials belong to the group whose ages are between 18 and 55. In that group, most of the subjects are male. On the contrary, for the age group whose ages are below 18, female subjects are more than males. The least represented age group among 1615 trials are whom ages above 55 and females dominate that group.

sEMG sampling frequency is important in determining preprocessing parameters, especially in filtering. The sampling frequencies of sEMG are not the same for all of the trials. Figure 2.3 shows the sEMG data which are sampled with 3 different frequencies. Among all trials, most of the signals are sampled at 960 Hz.

2.2 Algorithm development pipeline

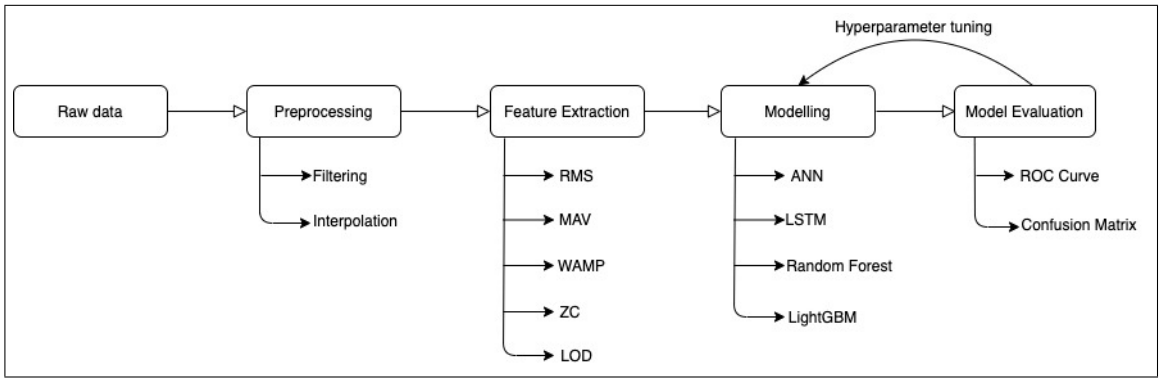


Figure 2.4 Flowchart of the feature engineering and modelling process.

Figure 2.4 shows the pipeline from raw data to performance evaluation through data preprocessing, feature engineering, and modelling. After the raw data were preprocessed by filtering, their lengths were equated by interpolation/extrapolation. Several widely used feature extraction techniques were used to extract meaningful information from the raw data before the modelling process. Classification results of the several machine learning algorithms were compared and their hyperparameters were tuned by measuring their performances with different metrics.

2.2.1 Preprocessing of sEMG data

Since the sEMG data's recording process is affected by external factors due to motion artifact, physiological or electromagnetic reasons during locomotion, the recorded raw data are prone to contain noise. The presence of noise may lead the machine learning algorithms not to perform their best performances or may lead to their failure during usage.

In this study, a band-pass filter was applied to the raw data to get rid of the noise. Figure 2.3 demonstrates that the sEMG data's sampling frequencies vary among 1615 trials, so the sampling frequency parameters in designing digital filters were specified by taking this variation into consideration. Thus, each subject's data were filtered individually.

After applying the preprocessing onto the raw data, the effects of the filtering were controlled by visual inspection for being sure that the frequency ranges and the order of the digital filter was appropriate. Figure 2.5 shows the comparison of raw and filtered sEMG Data of Subject 5, which behaves stochastically due to the noises in its raw form.

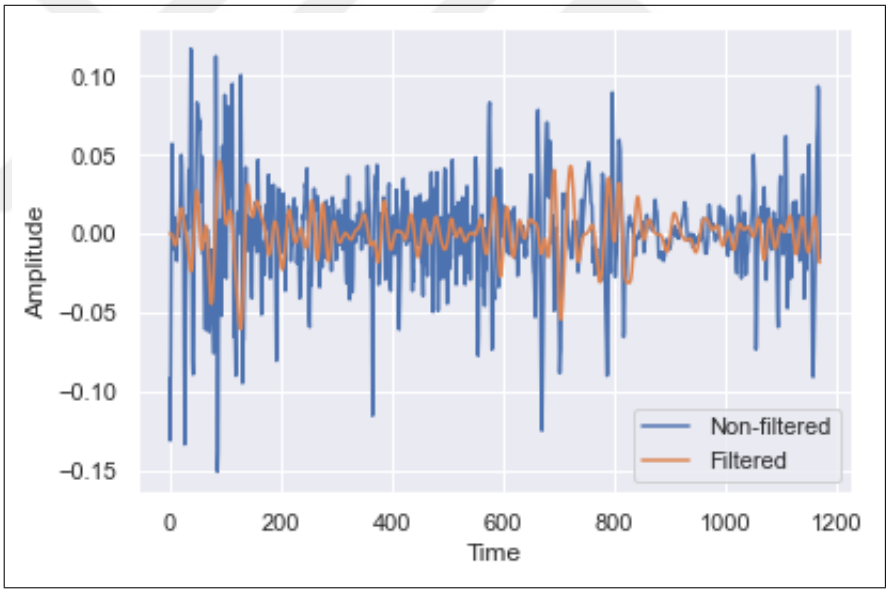


Figure 2.5 Comparison of raw and filtered sEMG Data of Subject 5.

Second-order Butterworth digital filter with a high-pass filter at 20 Hz and a low-pass filter at 200 Hz was designed. Figure 2.6 shows the performed Fast Fourier Analysis (FFT) for determining cutoff frequencies. The frequency domain was inferred from a time domain to determine the optimal cutoff frequencies. Because there is no deterministic way to determine the cutoff frequencies, FFT, which helps to bring significant frequency ranges into view, was used in this study. By visual inspection,

20-200 Hz frequency range was decided to be used in filtering.

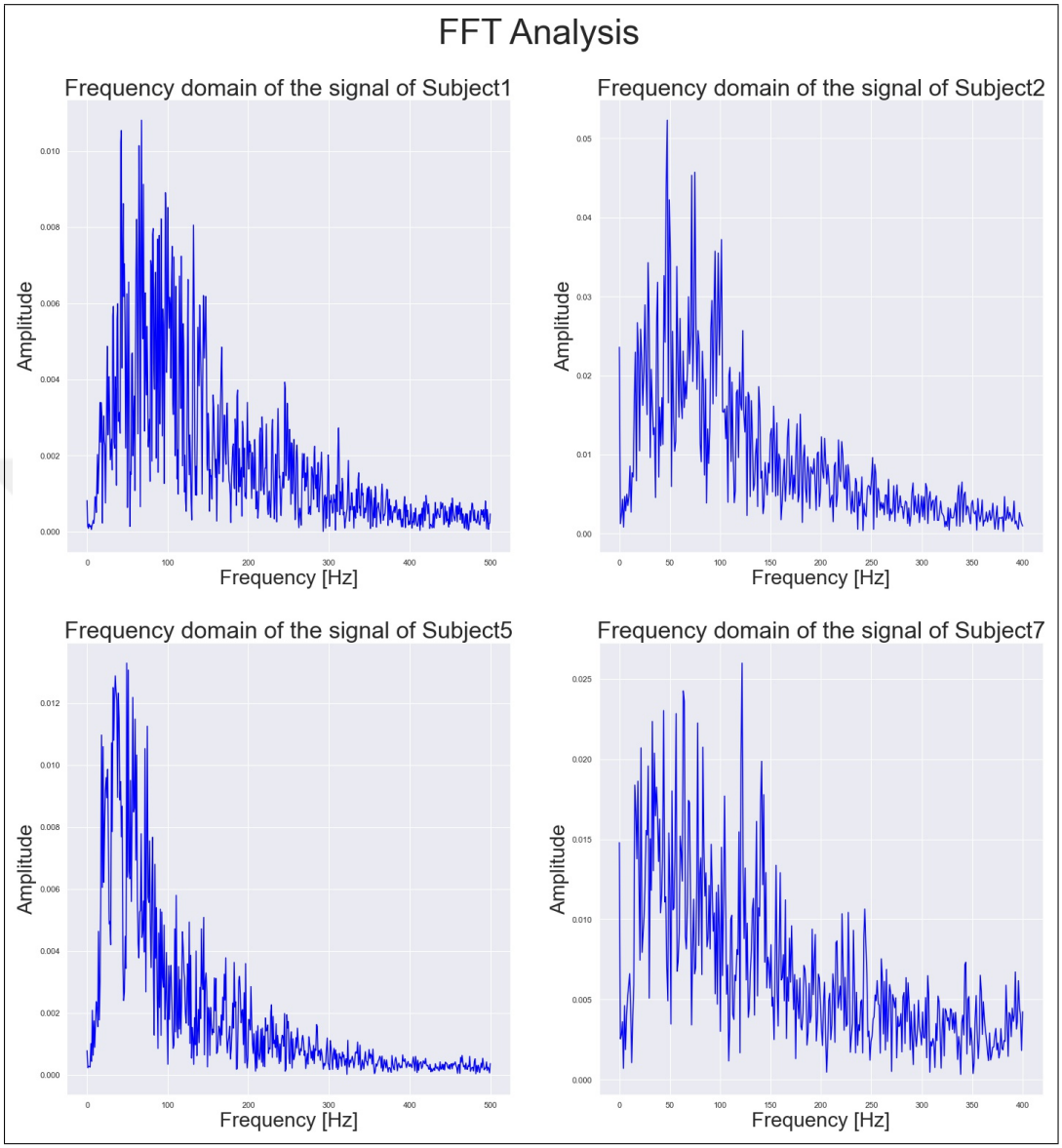


Figure 2.6 Selected Fast Fourier Analyses of sEMG data of Subjects 1, 2, 5 and 7.

2.2.2 Feature Extraction of sEMG data

After the noise was reduced by preprocessing the raw data, the following operation on the data before feeding into the models was feature extraction. In feature extraction, the main aim was to extract useful information from the raw data by mapping desired mathematical formulas onto the data. There are various formulas that are used in the literature.

Many of the widely used feature extraction algorithms were revised and some of them were tried even though they were not used in the final version of the classification of the sEMG data. Some of them were variance, v-order, and Slope Sign Changes.

Preliminary feature importance results and the correlation between independent features extraction algorithms conducted to decide final version of the features. All possible feature extraction algorithms could be used as there is no limit. However, as the number of features is increased, computational complexity is also increased. Thus, in this thesis carefully chosen five mathematically different algorithms were decided to be used. These algorithms are:

- 1) Willison Amplitude (WAM)
- 2) Root Mean Squared (RMS)
- 3) Mean Absolute Value (MAV)
- 4) Zero Crossing (ZCR)
- 5) Log Detector (LOD)

1) WAM is one of the best-performing feature extraction techniques for sEMG data, so it was decided to be included in this thesis. WAM can be defined as the number of times that the change in sEMG data amplitude exceeds a threshold ε; and it is calculated by the absolute value of the difference between sequential two-time stamps [51]. In this work, ε was taken as 0.005. WAM can also be an indication of the firing of motor unit action potentials because it has the capability to detect the instantaneous changes in a signal. Eq. 2.1 shows the mathematical formulation of WAM.

$$WAM = \sum_{i=1}^{N-1} f(|x_i - x_{i+1}|)$$
$$f(x) = \begin{cases} 1 & x > \varepsilon \\ 0 & otherwise \end{cases} \tag{2.1}$$

2) RMS is one of the most used feature extraction techniques in Machine Learning applications. RMS, or quadratic mean, can be defined as the total sum of the square root of each sEMG data in a single trial. It is calculated by squaring the values at each time stamp for whole data points and the results are summed up, then the results are divided by total data points in the trial. Finally, by taking the squared root of the value RMS value for the single-trial is obtained [51]. Eq. 2.2 shows the mathematical formulation of RMS.

$$RMS = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i)^2} \tag{2.2}$$

3) MAV is a widely used feature extraction technique in sEMG data classification as it contains information about sequential changes. In many ML models, MAV was used with both as a feature vector and as an error evaluator. MAV can be defined as the average of the absolute deviations between each data point and the mean of sEMG data in a single trial. After getting the average value for the whole data points of a single trial, the absolute difference between each data point value and overall mean is taken. Finally, MAV is obtained by dividing the sum of absolute differences of each data point by total data points [51]. Eq. 2.3 shows the mathematical formulation of MAV.

$$MAV = \frac{1}{N}\sum_{i=1}^{N}|x_i| \tag{2.3}$$

4) ZCR is one of the successful feature extraction techniques for signal processing and audio classification. Recently, there were some applications of it on sEMG data, so it was decided to be used in the sEMG data classification in this thesis. ZCR of the sEMG data can be defined as the number of times that the sEMG data amplitude crosses zero amplitude; in other words, the number of times the amplitude changes its sign [51]. Eq. 2.4 shows the mathematical formulation of ZCR.

$$\begin{aligned} ZCR &= \sum_{i=1}^{N-1} f(x_i) \\ \text{Where} \qquad f(x_i) &= \begin{cases} 1 & f(x_i) * f(x_{i+1}) < 0 \\ 0 & otherwise \end{cases} \end{aligned} \tag{2.4}$$

5) LOD is a feature extraction technique in sEMG data processing applications. LOD is a nonlinear feature extractor that helps to give an estimate of the exerted muscle force of the subjects. It is calculated by taking the logarithms of the absolute amplitudes at each data point for whole data points and the results are summed up, then the results are divided by total data points in the trial. Finally, the resultant to the exponent of e yields the LOD value for the single trial [51]. Eq. 2.5 shows the mathematical formulation of LOD.

$$LOD = \exp(\frac{1}{N}\sum_{i=1}^{N} log|x_i|) \tag{2.5}$$

2.2.3 Feature Selection of sEMG data

Normalizing the raw data, also our use case the sEMG data, is an affirmed operation in classification with the neural network-based models. However, even though normalization increases the performance, it also brings difficulties in real-time usage of powered ankle prostheses. The device will be used continuously by amputees in their daily lives, and they will perform plenty of various movements. When the signal gathered from amputees is normalized, the model may fail. For example, while sitting the muscles are not used even though the signals from the muscles continuously feed the classifier of the powered ankle prosthesis. If the normalization procedure is not used in the classifier's pipeline, the classifier can classify the movements correctly by taking relatively small amplitudes of the sEMG data. Contrarily, if normalization is performed, the tiny noise may be comprehended as a real muscle activation because of the scaling the features with respect to each other. Thus, even though the performance of classifiers using normalization is seemed as better theoretically, in real-time usage they are prone to fail.

Sliding window theorem provides continuous information from the raw signal and gives insights about changes occurring during movement because it processes sequential information as it moves. This windowing operation leads to having better accuracies, however, having more information comes with more computational complexity. In real-time usage of the powered ankle prostheses, it will affect the signal processing speed, the reaction of the device, and the battery consumption.

The ML algorithms which are planned to be used have different necessities in creating models. For example, while tree-based algorithms (LGBM and RF) do not need normalization due to the fact that they split the feature sets based on each features' ranges, on the other hand, for the neural network-based algorithms (LSTM and ANN) normalization is very crucial for performing their best. Non-normalizing the data has an adverse effect since they use back-propagation and gradient descent in their calculations.

On the other hand, these ML algorithms are also affected by windowing operations. Windowing increases computational complexity in constructing trees. Moreover, the tree-based algorithms draw their decision boundaries with respect to the variance in the data. Windowing operation causes these major variations to disappear since it causes smoothing the data. Thus windowing may lead tree-based models' accuracies to become worse. Conversely, LSTM's are specially designed with the intention of processing sequential data, so windowing operation provides more insightful information for the LSTM model while training.

With the purpose of optimizing the needs between computational complexity and having better performance, the selected models were planned to be trained and compared with different options. Because each of these algorithms has different needs for their best-performing cases, and of the limited time to complete the thesis, the responses of the models with respect to changing normalization and windowing operations were aimed to be compared. The four different feature set options are tabulated below:

- 1) Windowed and Normalized
- 2) Non-Windowed and Normalized
- 3) Windowed and Non-Normalized
- 4) Non-Windowed and Non-Normalized

The sEMG data from all of the eight muscles from the lower limb were planned to be used. For each of the eight muscles, after their raw sEMG data were preprocessed by filtering and inter/extrapolation, the desired features were extracted. All of the feature extraction algorithms explained in section 2.2.2 (WAM, RMS, MAV, ZC, LOD) were used in creating the four feature sets.

Among four feature selection sets, half of them were in different sizes due to windowing or non-windowing selections. In data pre-processing, each of the single trial's sEMG data sizes were reduced to 1000. After windowing, the data sizes were reduced by (window size -1). Window size was decided after trying several different options which were 10, 25,100, and 250. Eventually, the window size was determined as 50 to make a balance between computational complexity and sufficient information gathered from the data. For the Non-Normalized Windowed and Normalized Windowed feature sets, the data lengths became 951 (1000 - (50 -1) for each trial. For the Non-Normalized Non-Windowed and Normalized Non-Windowed feature sets, the data lengths were remained as before which was 1000. For each of these feature sets, the dimension which is the number of features obtained from the data was totally equal to 40 (8 muscles x 5 feature extraction algorithms).

2.3 Modelling of sEMG data

Four different algorithms from two different types of ML approaches were used in modelling. ANN and LSTM algorithms were used from the neural network-based algorithm approach. RF and LightGBM algorithms were used from the decision tree-based algorithm approach. This selection arised from the need to dissociate different strong and weak sides of these algorithms by changing different dataset selections.

Binary classification between Step-Up and Step-Down tasks was aimed to apply on the dataset. There was a total of 169 trials for the Step-Up tasks and 146 trials for Step-Down tasks in the dataset. 80% of the dataset were taken for training the model and 20% of the dataset were taken for testing the performance of the model. 130 trials of Step-Up and 118 trials of Step-Down tasks among 248 training data were used in training the model. 39 trials of Step-Up and 28 trials of Step-Down tasks among 67 test data were used in testing the performance of the classification model.

2.4 Postprocessing

2.4.1 Hyperparameter Selection

Hyperparameter selection was carried out via trying several options and selecting the best fitting option among them. For the sake of computational complexity, the RandomizedSearchCV function on the `sklearn.model_selection`module in the scikit-learn library was used to search for optimal hyperparameters for each of the models separately. GridSearchCV is another function of the `sklearn.model_selection`module and it scans all of the combinations of the parameters given, however, it is much more time consuming than RandomizedSearchCV. RandomizedSearchCV does not try all of the parameters given, instead, it decides parameters' importances while randomly trying. By picking the best possible options it reaches the optimal hyperparameters among the given range.

For four of the ML models used, the important parameters were determined, and possibly optimal hyperparameter ranges were decided separately. Some of the important parameters for tree-based ML models were the maximum depth of the tree, number of leaves in the tree, number of trees constructed, learning rate, and other parameters that determine the split of the features. Some of the important parameters for neural network-based models were optimizers, number of epochs, batch size, learning rate, etc. For both tree-based algorithms and neural network-based algorithms, regularization constant alpha and lambda, which respectively correspond to L1 and L2 regularization constants, have a crucial effect on preventing overfitting problems by trimming the weights given to the features.

2.4.2 Performance Evaluation

After the construction of the finalized binary classification models, each of the models assigned different probability values for each tasks while trying to predict each of the 67 test trials. Both of these probability values (the predicted step-up probability

and the predicted step-down probability) were between 0 and 1 and the summation of these two probability values was equal to 1. For predicting the task, each of the trials was assigned to the task which had a greater probability value than the other. Because the summation of these probability values were 1, a trial was classified as Step-Up task for the predicted step-up probability value with greater than 0.5. On the contrary, a trial was classified as Step-Down task for the predicted step-down probability value with greater than 0.5.

After each of the test trials were predicted, the performances of the classifications were assessed by comparing with the actual values. Confusion matrix is a table that compares and demonstrates the actual and the predicted labels. The table contains True-Positive, True-Negative, False-Positive and False-Negative prediction numbers. The accuracy values can be obtained by averaging the truly predicted trials over all the trials. The performances of the models across datasets are presented in the confusion tables.

Receiver Operating Characteristic (ROC) curve is a probability plot that demonstrates the performance of the binary classification model by taking True Positive Rate (TPR) and False Positive Rate (FPR) into consideration. Area Under the Curve (AUC) values demonstrate how well the classes are separated.

3. RESULTS

3.1 LightGBM Results

Tables 3.1, 3.2, 3.3, and 3.4 show the four confusion matrix for the results of the used LightGBM model. Table 3.4 shows the best test results of the LightGBM algorithm for binary classification between Step-Up and Step-Down tasks. Best result was obtained from Non-Windowed and Non-Normalized dataset.

Table 3.1
Confusion table of LightGBM model for Windowed and Normalized dataset.

		True task		
		Step-Up	Step-Down	Total
Predicted task	Step - Up	35	4	39
	Step - Down	4	24	28
	Total	39	28	67

Table 3.2
Confusion table of LightGBM model for Non-Windowed and Normalized dataset.

		True task		
		Step-Up	Step-Down	Total
Predicted task	Step - Up	34	1	35
	Step - Down	5	27	32
	Total	39	28	67

Table 3.3
Confusion table of LightGBM model for Windowed and Non-Normalized dataset.

		True task		
		Step-Up	Step-Down	Total
Predicted task	Step - Up	33	3	36
	Step - Down	6	25	31
	Total	39	28	67

Table 3.4
Confusion table of LightGBM model for Non-Windowed and Non-Normalized dataset.

		True task		
		Step-Up	Step-Down	Total
Predicted task	Step - Up	36	2	38
	Step - Down	3	26	29
	Total	39	28	67

3.2 LSTM Results

Tables 3.5, 3.6, 3.7, and 3.8 show the four confusion matrix for the results of the used LSTM model. Table 3.5 shows the best test results of the LSTM algorithm for binary classification between Step-Up and Step-Down tasks. Best result was obtained from Windowed and Normalized dataset.

Table 3.5
Confusion table of LSTM model for Windowed and Normalized dataset.

		True task		
		Step-Up	Step-Down	Total
Predicted task	Step - Up	36	1	37
	Step - Down	3	27	30
	Total	39	28	67

Table 3.6
Confusion table of LSTM model for Non-Windowed and Normalized dataset.

		True task		
		Step-Up	Step-Down	Total
Predicted task	Step - Up	35	2	37
	Step - Down	4	26	30
	Total	39	28	67

Table 3.7
Confusion table of LSTM model for Windowed and Non-Normalized dataset.

		True task		
		Step-Up	Step-Down	Total
Predicted task	Step - Up	35	3	38
	Step - Down	4	25	29
	Total	39	28	67

Table 3.8
Confusion table of LSTM model for Non-Windowed and Non-Normalized dataset.

		True task		
		Step-Up	Step-Down	Total
Predicted task	Step - Up	32	5	37
	Step - Down	7	23	30
	Total	39	28	67

3.3 RF Results

Tables 3.9, 3.10, 3.11, and 3.12 show the four confusion matrix for the results of the used RF model. Table 3.10 and 3.12 show the best test results of the RF algorithm for binary classification between Step-Up and Step-Down tasks. Best results were obtained from Non-Windowed and Normalized and Non-Windowed and Non-Normalized datasets.

Table 3.9
Confusion table of Random Forest model for Windowed and Normalized dataset.

		True task		
		Step-Up	Step-Down	Total
Predicted task	Step - Up	34	5	39
	Step - Down	5	23	28
	Total	39	28	67

Table 3.10
Confusion table of Random Forest model for Non-Windowed and Normalized dataset.

		True task		
		Step-Up	Step-Down	Total
Predicted task	Step - Up	35	4	39
	Step - Down	4	24	28
	Total	39	28	67

Table 3.11
Confusion table of Random Forest model for Windowed and Non-Normalized dataset.

		True task		
		Step-Up	Step-Down	Total
Predicted task	Step - Up	34	6	40
	Step - Down	5	22	27
	Total	39	28	67

Table 3.12
Confusion table of Random Forest model for Non-Windowed and Non-Normalized dataset.

		True task		
		Step-Up	Step-Down	Total
Predicted task	Step - Up	33	2	35
	Step - Down	6	26	32
	Total	39	28	67

3.4 ANN Results

Tables 3.13, 3.14, 3.15, and 3.16 show the four confusion matrix for the results of the used ANN model. Table 3.13 shows the best test results of the ANN algorithm for binary classification between Step-Up and Step-Down tasks. Best result was obtained from Windowed and Normalized dataset.

Table 3.13
Confusion table of ANN model for Windowed and Normalized dataset.

		True task		
		Step-Up	Step-Down	Total
Predicted task	Step - Up	36	6	42
	Step - Down	3	22	25
	Total	39	28	67

Table 3.14
Confusion table of ANN model for Non-Windowed and Normalized dataset.

		True task		
		Step-Up	Step-Down	Total
Predicted task	Step - Up	34	5	39
	Step - Down	5	23	28
	Total	39	28	67

Table 3.15
Confusion table of ANN model for Windowed and Non-Normalized dataset.

		True task		
		Step-Up	Step-Down	Total
Predicted task	Step - Up	29	4	33
	Step - Down	10	24	34
	Total	39	28	67

Table 3.16
Confusion table of ANN model for Non-Windowed and Non-Normalized dataset.

		True task		
		Step-Up	Step-Down	Total
Predicted task	Step - Up	27	3	30
	Step - Down	12	25	37
	Total	39	28	67

3.5 Best performing algorithm selection

The results were obtained from the test dataset with size 67. Table 3.17 shows all of the test results of the binary classification between Step-Up and Step-Down tasks for all models. Tables 3.4, 3.5, 3.10, 3.12 and 3.13 show the best performing case results of the used models. Figure 3.1 shows the ROC curve for all of the 4 models with their best performing cases with their Area Under the Curve (AUC) values. The best accuracy and AUC value were obtained by LSTM from the Windowed and Normalized dataset. The best Non-Normalized accuracy and AUC value were obtained by LightGBM from the Non-Windowed and Non-Normalized dataset.

Table 3.17
Accuracy percentages of four models with respect to changing feature extraction methods.

	LightGBM	LSTM	RF	ANN
Windowed and Normalized	88.05%	94.02%	85.07%	86.56%
Non-Windowed and Normalized	91.04%	91.04%	88.05%	85.07%
Windowed and Non-Normalized	86.56%	89.55%	83.58%	79.10%
Non-Windowed and Non-Normalized	92.53%	82.08%	88.05%	77.61%
Average accuracy percentages:	89.55%	89.17%	86.19%	82.09%

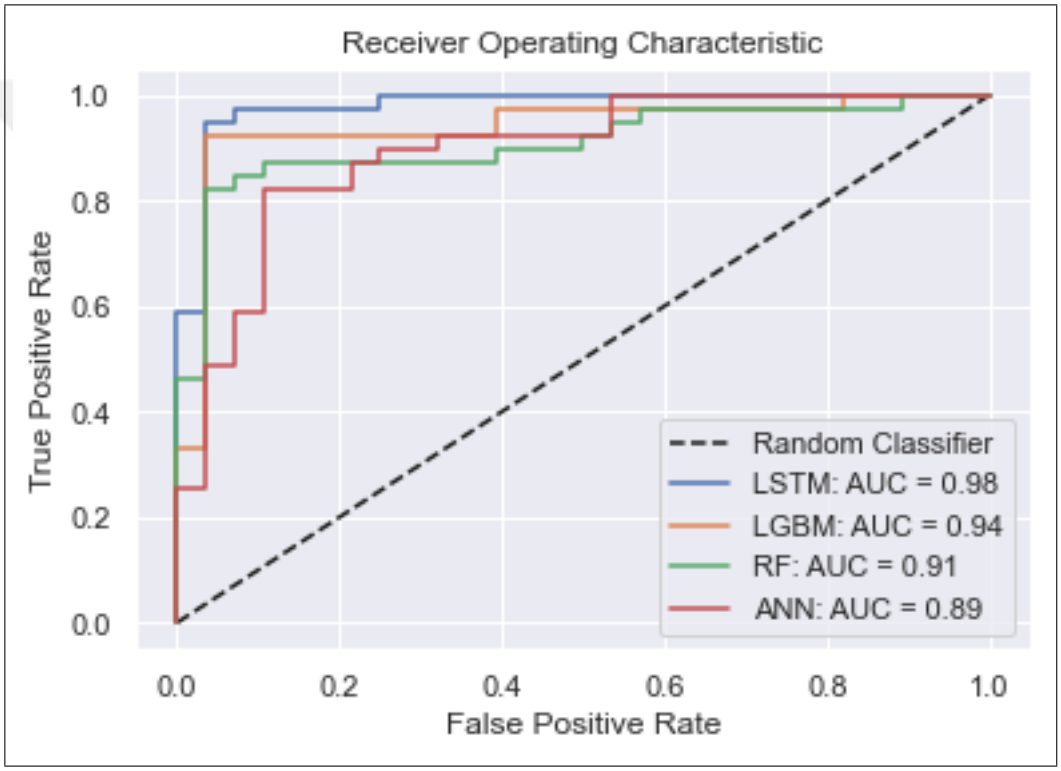


Figure 3.1 ROC curve of the classification results of all models with the best performing feature extraction selections.

4. DISCUSSION

4.1 Comparison of the classification algorithms

Feature selection preferences and different models have an impact on the performance of the classification. All of the four feature selection algorithms have advantages and disadvantages individually. The main goal was to classify two tasks while constructing a less complex model which also meets the needs of the real-time device without making concessions on accuracy. Thus, four different ML models with the four different feature set from the same dataset were used and the needs of the powered ankle prosthesis are tried to be optimized.

Among four different ML models and 4 different feature sets, the best accuracy score is 94.02% with the best AUC value of 98% which is obtained by the LSTM model for the Windowed and Normalized dataset. The second-best accuracy is obtained by the LightGBM model for the Non-Windowed and Non-Normalized dataset with an accuracy score of 92.53% and AUC value of 94%. Random Forest and ANN models also accurately predict the Step-Up and Step-Down tasks, but LightGBM and LSTM models showed outstanding performance.

LSTM model's performance is drastically decreased in Non-Windowed and Non-Normalized dataset. The average accuracy of the LSTM model is 89.17%, while the best accuracy score of LSTM is 94.02% with the best AUC value of 98%. The first main reason is that non-normalization had an adverse impact on the performance of neural network-based algorithms, which also leads to a 4.5% difference in accuracy score between two windowed cases. The second main reason can be noticed by the 7.5% difference between two Non-Normalized cases which implies that the windowing is also an important operation that helps to have better performing LSTM model, as the windowing operation provides more sequential information and the LSTM model have a specialized design to cope with the sequential data.

Thus, the variation in the results for the LSTM model is not surprising as it was known that the neural network-based algorithms are in need of normalization and windowing operation.

LightGBM model outperforms in the classification of the two tasks, it has the highest average accuracy percentage with 89.55%. Its best performance is observed in Non-Windowed and Non-Normalized dataset and in the Non-Windowed and Normalized dataset there observed only one extra wrong prediction, so it can also be seen as another optimal choice. Comparing the results of LightGBM for two windowed datasets, it can be seen that the Normalized dataset yielded better performance. Comparing the two Normalized and two Non-Normalized datasets in themselves shows that in the Non-Windowed dataset, the performance is turned out to be better. Thus, it should be considered from the thesis's results that, for having the best performing LightGBM model, the windowing operation should not be performed and the features should be processed as a whole.

Random Forest model's results are worse than LSTM and LightGBM, however, in the Non-Normalized and Non-Windowed case, its performance is better than LSTM. Similar to the LightGBM model, the windowing operation also worsened the accuracy results and the best performances of it are delivered in Non-Windowed cases with the same accuracies of 88.05% with an AUC value of 91%. Normalization has no significant impact on the performance of the Random Forest model as the tree-based algorithms do not need it. The distinctive reason that causes a difference in accuracy results between the Random Forest model and the LightGBM model is Gradient Boosting Machine that LightGBM uses, which has the ability to increase the model's performance.

ANN model's results are the worst among the four models with an average accuracy of 82.09% and an AUC score of 89%. Similar to the LSTM model's results, it performed significantly worse in the Non-Normalized datasets due to the nature of neural network-based algorithms. Windowing operation seemed to have an increasing impact on performance, but it has not had a huge impact on it.

Due to the specialized design in processing sequential data, a higher performance is expected from the LSTM model than ANN for classifying sequential data like sEMG data, as the results suggest. There might be other reasons resulting in the accuracy gap between the two models being that high, like the inappropriate choice of feature selection algorithms and insufficient hyperparameter tuning. However, in the ideal cases for both of the models, LSTM is expected to have better performance than ANN.

Tree-based algorithms are more robust than the dataset choice changes than neural-network-based algorithms. Especially in Non-Windowed and Non-Normalized cases, tree-based models performed much better than the neural network-based models. Variation in the average accuracies is higher in ANN and LSTM than LightGBM and Random Forest. However, for the Windowed and Non-Normalized dataset, LSTM has the best accuracy among the four models even if normalization is vital for LSTM to perform its best. The results for this dataset choice separates the requirements of two types of ML algorithms: The neural network-based algorithms require normalized data and the tree-based algorithms require non-duplicated and varying data

. Training data fed to the deep neural networks (DNN) like ANN and LSTM has a significant effect on their performance. When the feature set is not of good quality or does not contain distinctive information about the target trying to be predicted, deep neural networks might fail to configure their model weights to predict the outcome values correlatively with the true outcome value. In the thesis, five proven success feature extraction algorithms are used to create feature sets without including the raw data itself. These features might not be predictive as much as preceding use cases in the literature, however investigating this possibility goes beyond the purpose of this thesis.

This thesis aimed to classify Step-Up tasks from Step-Down tasks without normalization as much successful as the normalized cases. Because there are plenty of accurately performing classification models in the literature, the aim is to have closer accuracy scores without doing normalization. As a result, even though the LSTM model with normalized case showed the best accuracy among 16 results, LightGBM with the non-normalized case also showed very close results to the LSTM model with 1.5% difference, which corresponds to an extra of one true prediction, that can be tolerable.

Additionally, the best-performing feature set of LightGBM is a Non-Windowed case which results in designing a cost-efficient device in terms of both computational complexity, device response, and the battery. Thus, even though the accuracy of the LightGBM model is slightly lower than the LSTM model, the advantages of using it make this accuracy difference negligible. After switching to a high-performance computer that can be used as a training platform, wider hyperparameter ranges can be tried and the accuracy scores can be improved. Therefore, in the final version of Powered Ankle Prosthesis, it is appropriate to use the LightGBM model in the classification of Step-Up and Step-Down tasks regarding its advantage to accurately classify non-normalized data with cost-efficient feature extraction techniques.

4.2 Future Work

There are some successful implications of processing non-normalized sEMG data in the literature. Accurately predicting the position and the moment of the ankle with a regression model made me work to develop a classification model with the aim of processing the non-normalized sEMG data to accurately implement the classification version of these successful models. In this thesis, among the four datasets created, half of them contained non-normalized datasets with the purpose of understanding how the performance of the classifier differs when not normalizing the data.

It is seen from the results of the Random Forest and LightGBM models that classifiers' performance is not affected significantly when the data is not normalized. Therefore, I will look for innovative ways to improve the accuracy scores for these tree-based models. Since the results showed the ineffectiveness of the normalization procedure in the tree-based models, I will create my future models without including normalization confidently.

Five different commonly used feature extraction algorithms are used in the thesis, however, there are many different algorithms that can be used in creating the training set. More algorithms to extract features from the raw data results in having higher dimensional data which increases computational complexity. Together with the learnings from this thesis, one of my future aims is to develop the classification model by using extra different feature extraction algorithms in addition to the algorithms used in this thesis.

DNNs like ANN and LSTM have an advantageous property that can be used in classifying sEMG data. They can process the raw data without extracting features explicitly [46]. There are different success proven use cases of DNN in image and speech classifications. This advantage arises due to the fact that DNN can extract useful information at the same time it does the training. Feature extraction in DNN is done within the hidden layers in detail which causes almost perfect predictions by correlating these details with the targets, however, the features cannot be interpretable since it is created and consumed in the hidden layers. As another next step, with the purpose of comparing the results of this thesis, ANN and LSTM will be tried with their raw forms for the same dataset used in this thesis.

Hyperparameters of the ML models can significantly influence the models' performances. Because of having limited time and computational resources, hyperparameter selections might not be optimized by searching the optimal hyperparameters in a narrow range. In the following works wider hyperparameter ranges will be searched and more optimal hyperparameter values will be obtained.

There are a total of 315 trials in the dataset for the Step-Up and Step-Down tasks. 248 of them were used in training the models and the rest were used for testing. The number of trials might be insufficient to develop well-performing models. Therefore, either by obtaining from the literature or by recording in our laboratory in the future, the ML models will be developed by using a bigger dataset to develop better-performing classifiers.

Even though there are five different tasks available in the used dataset, binary classification between Step-Up and Step-Down tasks are made in this work. There are approximately 1300 more data available that are not used in developing classification models. Half of the dataset is for the walking tasks with changing speeds. As an upcoming task, I will try to develop a multi-class classification model in which all of the five tasks will be classified separately. Because the trial numbers are variant (especially walking tasks) among the tasks, the dataset is imbalanced. Thus, developing a two-step model which firstly predicts the task is walking or not followed by a multi-class classification among the remaining four tasks might be implemented to overcome the imbalanced dataset problem.

There are different advantages and disadvantages of the used models. A trial's task can be predicted wrong by a model, on the other hand, another model can correctly predict the same trial. The models assigns probability values for each task to predict the tasks. A false prediction can be made with a 51% probability value, whereas another model might predict the same trial correctly with a probability value of 80%. In future works, an optimized and reliable way to fuse model predictions will be searched to have more accurate classifiers.

Keles et al. used 5 muscles to successfully develop control algorithms for ground level walking task [11]. In this thesis, I used 8 different muscles and some of them may be unnecessary. Because there are different kinds of ampuations in the lower limb, some of these muscles cannot be available in every amputee. Thus, I will look for the ways to develop this thesis's work by taking 5 or less muscles as input to have similar accuracies eventually.

I have done binary classification in this thesis for classifying Step-Up and Step-Down tasks. In the following works, I aim to make multiclass classification between Step-Up, Step-Down and Walking tasks. This will make classification control algorithm of the Powered Ankle Prosthesis more developed.

To sum up, I will try to find a higher performance computing environment or a cloud environment over a longer period of time in order to achieve all of the future goals. When these goals are achieved, through the experience and knowledge gained, I believe we will be more closer to implement these smart technologies onto the Powered Ankle Prostheses.

5. CONCLUSION

The aim of this thesis is to develop control algorithms for Powered Ankle Prostheses to classify stair ascending tasks by using sEMG Data and ML techniques. Before testing on the amputees, the data collected from 50 healthy subjects were used to develop binary classification algorithms for Step-Up and Step-Down tasks. To maintain an efficient algorithm, obstacles in the daily usage of the prostheses are taken into account, and the development path is drawn accordingly. Different perspectives in processing the sEMG data are reviewed in the literature and the methods of this thesis are developed to investigate the different options mitigating the disadvantages of using the real-time classification algorithms in the Powered Ankle Prostheses.

One of the problems occurring in the real-time usage of sEMG data classification arises from the normalization of the sEMG data during processing. Normalization of the signal can result in processing misleading information, for example, while not making any movement any noise can be comprehended as a significant amplitude which causes control algorithms to fail. Another problem is the usage of the sliding window theorem during extracting the features which is computationally expensive. To investigate the benefits of using normalization and windowing, four different feature sets are created by inclusive and exclusive combinations of them. Also, for understanding model-based effects of them, four different ML algorithms originated from two different methodologies are used.

The results suggest that tree-based algorithms, especially LightGBM, performed very well in the Non-Normalized and Non-Windowed dataset whose usage in real-time prostheses helps to mitigate the disadvantages. On the other hand, the best accuracy percentage and the AUC value are obtained from the LSTM model for the Windowed and Normalized dataset which might be problematic in real-time usage. To sum up, by focusing on non-normalized datasets, I will look through the more accurate and reliable ways of designing control algorithms for powered ankle prostheses in the future.

REFERENCES

1. Hall, S. J., M. Provost-Craig, and W. C. Rose, *Chapter 5.3: The Major Skeletal Muscles*, p. 180. The Goodheart-Willcox Company, Inc., 2014.

2. Eilenberg, M. F., H. Geyer, and H. Herr, "Control of a powered ankle foot prosthesis based on a neuromuscular model," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, Vol. 18, no. 2, pp. 164–173, 2010.

3. Arnold, E. M., S. R. Ward, R. L. Lieber, and S. L. Delp, "A model of the lower limb for analysis of human movement," *Annals of Biomedical Engineering*, Vol. 38, no. 2, pp. 269–279, 2009.

4. Juneja, P., and J. B. Hubbard, *Anatomy, Bony Pelvis and Lower Limb, Tibialis Anterior Muscles*, StatPearls Publishing, Treasure Island (FL), 2020.

5. Drake, R. L., W. Vogl, A. W. M. Mitchell, R. Tibbitts, P. Richardson, A. Horn, and H. Gray, *Gray's Atlas Of Anatomy*, Elsevier, 2021.

6. Palastanga, N., and R. Soames, *Anatomy And Human Movement*, Churchill Livingstone, 2012.

7. Windrich, M., M. Grimmer, O. Christ, S. Rinderknecht, and P. Beckerle, "Active lower limb prosthetics: A systematic review of design issues and solutions," *BioMedical Engineering Online*, Vol. 15, 2016.

8. Lenzi, T., M. Cempini, L. Hargrove, and T. Kuiken, "Design, development, and testing of a lightweight hybrid robotic knee prosthesis," *International Journal of Robotics Research*, Vol. 37, pp. 953–976, jul 2018.

9. Au, S. K., and H. M. Herr, "Powered ankle-foot prosthesis," *IEEE Robotics and Automation Magazine*, Vol. 15, no. 3, pp. 52–59, 2008.

10. Borjian, R., J. Lim, M. B. Khamesee, and W. Melek, "The design of an intelligent mechanical active prosthetic knee," in *2008 34th Annual Conference of IEEE Industrial Electronics*, pp. 3016–3021, 2008.

11. Keles, A. D., and C. A. Yucesoy, "Development of a neural network based control algorithm for powered ankle prosthesis," *Journal of Biomechanics*, Vol. 113, p. 110087, 2020.

12. Herr, H., "Exoskeletons and orthoses: Classification, design challenges and future directions," *Journal of NeuroEngineering and Rehabilitation*, Vol. 6, no. 1, 2009.

13. Grimmer, M., *Powered Lower Limb Prostheses*. PhD thesis, Technical University Darmstadt, 2015.

14. Hu, B., A. M. Simon, and L. Hargrove, "Deep Generative Models With Data Augmentation to Learn Robust Representations of Movement Intention for Powered Leg Prostheses," *IEEE Transactions on Medical Robotics and Bionics*, Vol. 1, pp. 267–278, nov 2019.

15. Tan, H. X., N. N. Aung, J. Tian, M. C. H. Chua, and Y. O. Yang, "Time series classification using a modified LSTM approach from accelerometer-based data: A comparative study for gait cycle detection," *Gait and Posture*, Vol. 74, pp. 128–134, oct 2019.

16. Atzori, M., M. Cognolato, and H. Müller, "Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands," *Frontiers in Neurorobotics*, Vol. 10, no. SEP, 2016.

17. Novak, A., and B. Brouwer, "Sagittal and frontal lower limb joint moments during stair ascent and descent in young and older adults," *Gait & Posture*, Vol. 33, pp. 54–60, jan 2011.

18. Faisal, A. I., S. Majumder, T. Mondal, D. Cowan, S. Naseh, and M. J. Deen, "Monitoring methods of human body joints: State-of-the-art and research challenges," *Sensors*, Vol. 19, jun 2019.

19. Horst, F., S. Lapuschkin, W. Samek, K. R. Müller, and W. I. Schöllhorn, "Explaining the unique nature of individual gait patterns with deep learning," *Scientific Reports*, Vol. 9, dec 2019.

20. Thatte, N., T. Shah, and H. Geyer, "Robust and adaptive lower limb prosthesis stance control via extended kalman filter-based gait phase estimation," *IEEE Robotics and Automation Letters*, Vol. 4, pp. 3129–3136, oct 2019.

21. Morbidoni, C., A. Cucchiarelli, S. Fioretti, and F. Di Nardo, "A Deep Learning Approach to EMG-Based Classification of Gait Phases during Level Ground Walking," *Electronics*, Vol. 8, no. 8, 2019.

22. Bovi, G., M. Rabuffetti, P. Mazzoleni, and M. Ferrarin, "A multiple-task gait analysis approach: Kinematic, kinetic and EMG reference data for healthy young and adult subjects," *Gait and Posture*, Vol. 33, p. 6, Jan 2011.

23. Kumar, P., S. Mukherjee, R. Saini, P. Kaushik, P. P. Roy, and D. P. Dogra, "Multimodal gait recognition with inertial sensor data and video using evolutionary algorithm," *IEEE Transactions on Fuzzy Systems*, Vol. 27, no. 5, pp. 956–965, 2019.

24. Alharthi, A. S., S. U. Yunas, and K. B. Ozanyan, "Deep learning for monitoring of human gait: A review," *IEEE Sensors Journal*, Vol. 19, pp. 9575–9591, nov 2019.

25. Jung, J. Y., W. Heo, H. Yang, and H. Park, "A neural network-based gait phase classification method using sensors equipped on lower limb exoskeleton robots," *Sensors (Switzerland)*, Vol. 15, pp. 27738–27759, oct 2015.

26. Mercer, V. S., M. T. Gross, S. Sharma, and E. Weeks, "Comparison of gluteus medius muscle electromyographic activity during forward and lateral step-up exercises in older adults," *Physical Therapy*, Vol. 89, pp. 1205–1214, Nov 2009.

27. Toledo-Perez, D. C., M. A. Martinez-Prado, R. A. Gomez-Loenzo, W. J. Paredes-Garcia, and J. Rodriguez-Resendiz, "A study of movement classification of the lower limb based on up to 4-emg channels," *Electronics*, Vol. 8, no. 3, 2019.

28. Pires, R., T. Falcari, A. B. Campo, B. C. Pulcineli, J. Hamill, and U. F. Ervilha, "Using a support vector machine algorithm to classify lower-extremity emg signals during running shod/unshod with different foot strike patterns," *Journal of Applied Biomechanics*, Vol. 35, pp. 87–90, Feb 2019.

29. Xia, P., J. Hu, and Y. Peng, "Emg-based estimation of limb movement using deep learning with recurrent convolutional neural networks," *Artificial Organs*, Vol. 42, no. 5, pp. E67–E77, 2018.

30. Fleischer, C., and G. Hommel, "A human–exoskeleton interface utilizing electromyography," *IEEE Transactions on Robotics*, Vol. 24, no. 4, pp. 872–882, 2008.

31. McManus, L., G. De Vito, and M. M. Lowery, "Analysis and biophysics of surface emg for physiotherapists and kinesiologists: Toward a common language with rehabilitation engineers," *Frontiers in Neurology*, Vol. 11, p. 1216, 2020.

32. Patil, N. M., and S. R. Patil, "Review on real-time emg acquisition and hand gesture recognition system," in *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, Vol. 1, pp. 694–696, 2017.

33. Fleming, A., N. Stafford, S. Huang, X. Hu, D. P. Ferris, and H. H. Huang, "Myoelectric control of robotic lower limb prostheses: A review of electromyography interfaces, control paradigms, challenges and future directions," *Journal of Neural Engineering*, Vol. 18, no. 4, p. 041004, 2021.

34. Emanuel Singh, R., K. Iqbal, G. White, and J. K. Holtz, "A Review of EMG Techniques for Detection of Gait Disorders," *Artificial Intelligence - Applications in Medicine and Biology*, Vol. 1, no. 1, 2019.

35. Chowdhury, R. H., M. B. Reaz, M. A. Bin Mohd Ali, A. A. Bakar, K. Chellappan, and T. G. Chang, "Surface electromyography signal processing and classification techniques," *Sensors*, Vol. 13, pp. 12431–12466, sep 2013.

36. Novak, D., and R. Riener, "A survey of sensor fusion methods in wearable robotics," in *Robotics and Autonomous Systems*, Vol. 73, pp. 155–170, Elsevier, nov 2015.

37. Reaz, M. B., M. S. Hussain, and F. Mohd-Yasin, "Techniques of EMG signal analysis: Detection, processing, classification and applications," *Biological Procedures Online*, Vol. 8, pp. 11–35, mar 2006.

38. Nazmi, N., M. A. A. Rahman, S. A. Mazlan, H. Zamzuri, and M. Mizukawa, "Electromyography (emg) based signal analysis for physiological device application in lower limb rehabilitation," in *2015 2nd International Conference on Biomedical Engineering (ICoBE)*, pp. 1–6, 2015.

39. Zhang, X., Y. Wang, and R. P. Han, "Wavelet transform theory and its application in emg signal processing," in *2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery*, Vol. 5, pp. 2234–2238, 2010.

40. Tuncer, T., S. Dogan, and A. Subasi, "Surface emg signal classification using ternary pattern and discrete wavelet transform based feature extraction for hand movement recognition," *Biomedical Signal Processing and Control*, Vol. 58, p. 101872, 2020.

41. "Fast fourier transform." `https://en.wikipedia.org/wiki/Fast_Fourier_transform`, Aug 2021. Accessed: 2021-08-08.

42. Roland, T., S. Amsuess, M. F. Russold, and W. Baumgartner, "Ultra-low-power digital filtering for insulated emg sensing," *Sensors*, Vol. 19, no. 4, 2019.

43. "Butterworth filter." `https://en.wikipedia.org/wiki/Butterworth_filter`, Aug 2021. Accessed: 2021-08-08.

44. Mello, R. G., L. F. Oliveira, and J. Nadal, "Digital butterworth filter for subtracting noise from low magnitude surface electromyogram," *Computer Methods and Programs in Biomedicine*, Vol. 87, no. 1, pp. 28–35, 2007.

45. Wahid, M. F., R. Tafreshi, M. Al-Sowaidi, and R. Langari, "Subject-independent hand gesture recognition using normalization and machine learning algorithms," *Journal of Computational Science*, Vol. 27, pp. 69–76, 2018.

46. Jabbari, M., R. N. Khushaba, and K. Nazarpour, "Emg-based hand gesture classification with long short-term memory deep recurrent neural networks," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, pp. 3302–3305, 2020.

47. Halaki, M., and K. Gi, "Normalization of emg signals: To normalize or not to normalize and what to normalize to?," *Computational Intelligence in Electromyography Analysis - A Perspective on Current Applications and Future Challenges*, 2012.

48. Emayavaramban, G., A. Amudha, M. S. Ramkumar, S. Divyapriya, and P. Nagaveni, "Surface eletromyography feature extraction techniques – a review," in *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 441–446, 2020.

49. Hudgins, B., P. Parker, and R. Scott, "A new strategy for multifunction myoelectric control," *IEEE Transactions on Biomedical Engineering*, Vol. 40, no. 1, pp. 82–94, 1993.

50. Rechy, E., and H. Hu, "Bio-signal based control in assistive robots: A survey," *Digital Communications and Networks*, Vol. 2, 03 2015.

51. Phinyomark, A., P. Phukpattaranont, and C. Limsakul, "Feature reduction and selection for emg signal classification," *Expert Systems with Applications*, Vol. 39, no. 8, pp. 7420–7431, 2012.

52. Mukhopadhyay, A. K., and S. Samui, "An experimental study on upper limb position invariant emg signal classification based on deep neural network," *Biomedical Signal Processing and Control*, Vol. 55, p. 101669, 2020.

53. Liu, J., and P. Zhou, "A novel myoelectric pattern recognition strategy for hand function restoration after incomplete cervical spinal cord injury," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, Vol. 21, no. 1, pp. 96–103, 2013.

54. Stango, A., F. Negro, and D. Farina, "Spatial correlation of high density emg signals provides features robust to electrode number and shift in pattern recognition for myocontrol," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, Vol. 23, no. 2, pp. 189–198, 2015.

55. Tavakoli, M., C. Benussi, P. A. Lopes, L. B. Osorio, and A. T. de Almeida, "Robust hand gesture recognition with a double channel surface emg wearable armband and svm classifier," *Biomedical Signal Processing and Control*, Vol. 46, pp. 121–130, 2018.

56. Lian, K.-Y., C.-C. Chiu, Y.-J. Hong, and W.-T. Sung, "Wearable armband for real time hand gesture recognition," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2992–2995, 2017.

57. Cote-Allard, U., C. L. Fall, A. Drouin, A. Campeau-Lecours, C. Gosselin, K. Glette, F. Laviolette, and B. Gosselin, "Deep learning for electromyographic hand gesture signal classification using transfer learning," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, Vol. 27, no. 4, pp. 760–771, 2019.

58. Lu, Y., H. Wang, Y. Qi, and H. Xi, "Evaluation of classification performance in human lower limb jump phases of signal correlation information and lstm models," *Biomedical Signal Processing and Control*, Vol. 64, p. 102279, 2021.

59. Wang, S.-C., "Artificial neural network," *Interdisciplinary Computing in Java Programming*, pp. 81–100, 2003.

60. Shahkarami, A., S. D. Mohaghegh, V. Gholami, and S. A. Haghighat, "Artificial intelligence (ai) assisted history matching," *All Days*, 2014.

61. Hochreiter, S., and J. Schmidhuber, "Long short-term memory," *Neural Computation*, Vol. 9, no. 8, pp. 1735–1780, 1997.

62. Mozer, M. C., "Induction of multiscale temporal structure," in *Proceedings of the 4th International Conference on Neural Information Processing Systems*, NIPS'91, (San Francisco, CA, USA), pp. 275–282, Morgan Kaufmann Publishers Inc., 1991.

63. Olah, C., "Understanding lstm networks." `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`, May 2015. Accessed: 2021-06-05.

64. Education, I. C., "What is random forest?." `https://www.ibm.com/cloud/learn/random-forest`. Accessed: 2021-06-15.

65. Education, I. C., "What is boosting?." `https://www.ibm.com/cloud/learn/boosting`. Accessed: 2021-06-15.

66. Microsoft, "Lightgbm's documentation." `https://lightgbm.readthedocs.io/en/latest/`. Accessed: 2021-07-01.

67. Microsoft, "Features of lightgbm." `https://lightgbm.readthedocs.io/en/latest/Features.html`. Accessed: 2021-07-01.

68. Lencioni, T., I. Carpinella, M. Rabuffetti, A. Marzegan, and M. Ferrarin, "Human kinematic, kinetic and EMG data during different walking and stair ascending and descending tasks," *Scientific Data*, Vol. 6, no. 1, p. 309, 2019.