ADVANCED DEEP LEARNING APPROACHES FOR BIOSIGNALS APPLICATIONS

By Marwa Farouk Ibrahim Ibrahim

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

PhD in the School of Biomedical Engineering, Faculty of Engineering and Information Technology
The University of Technology Sydney (UTS)

January 2019

Certificate of Original Authorship

I, Marwa Farouk Ibrahim, certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged.

In addition, I certify that all information sources and literature used are indicated in the thesis.

This research is supported by the Australian Government Research Training Program.

Signature of Student: Production Note:
Signature removed prior to publication.

Date: 11 January 2019

Acknowledgment

I would like to express my deep gratitude and extend my warmest and most sincere thanks to my co-supervisor **Associate Professor Robert Fitch** for all his continous support, efforts, endless help, guidance, understanding and discussions.

I would like to thank **Dr Nham Tran** for his generous support which otherwise I will not be able to finish my thesis.

I would like also to thank **Associate Professor Gyorgy Hutvagner** for his powerful decision that helped me finish my thesis.

I would like to extend further my genuine appreciations to **Dr Ben Rodanski** for giving me the chance of working as a casual academic at UTS.

I would also like to thank **Dr. Khairul Anam** who generously supported me with the EMG datasets and valuable discussions.

I would like to pass my thanks to **Associate Professor Adel Al- Jumaily** for the datasets and resources that he provided me.

Last but not least, I would like to express my special gratitude to my beloved father **Farouk**, wounderful mother **Faten**, lovely brothers **Hesham** and **Emad** and my mother-in-law **Nour El Hoda** for their continuous support that motivated me to finish my thesis.

A very special thanks goes to my beloved husband, **Tawfek Mahmoud**, for his endless support, love, encouragement, motivation, and patience during my entire years of study.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ANN	Artificial neural network
ANOVA	Analysis of variance
AR	Autoregressive
AW-ELM	Adaptive wavelet extreme learning machine
CNN	Convolutional neural network
CDF	Cumulative distribution function
CMC	Carpometacarpal
CPD	Canonical polyadic decomposition
CTC	Connectionist temporal classification
DA	Discriminant analysis
DCS	Dynamic classifier selection
DCS-LA	Dynamic classifier selection local accuracy
DFT	Discrete Fourier transform
DTNN	Deep tensor neural network
ECG	Electrocardiogram
ECOG	Electrocorticographic
ECRB	Extensor carpi radialis brevis
ECRL	Extensor carpi radialis longus
ECU	Extensor carpi ulnaris
EDA	Electrodermal activity
EEG	Electroencephalogram
ELM	Extreme learning machine
EMG	Electromyography
EOG	Electrooculography
FCU	Flexor carpi ulnaris
FFT	Fast Fourier transform

FLDA	Fuzzy LDA
FLN	Functional-link net
FNN	Feed-forward neural networks
FNPA	Fuzzy neighbourhood preserving analysis
FP	Feature projection
FS	Feature selection
FT	Fourier transform
GSR	Galvanic skin response
HME	Hierarchical mixture of experts
HMM	Hidden Markov model
HTD	Hjorth time domain
ICA	Independent component analysis
KNN	k nearest neighbourhood
K-SVD	K-singular value decomposition
LDA	Linear discriminant analysis
LIBSVM	Library support vector machine
LinELM	Linear extreme learning machine
LLE	Local linear embedding
LPM	Linear programming machine
LPP	Locality preserving projection
LS-SVM	Least-square SVM
LSTM	Long Short-Term Memory
MAV	Mean absolute value
MAVs	Mean absolute value slope
MC	Markov chain
MCA	Mutual components analysis
MCE	Misclassification error
MCMC	Markov chain Monte Carlo

MDF	Median frequency
MDP	Markov decision process
MEG	Magnetoencephalogram
MES	Myoelectric signal
MI	Mutual information
MLP	Multilayer perceptron
MMG	Mechanomyogram
MNF	Mean frequency
NB	Naive bayes
NMF	Nonnegative matrix Factorization
OFNDA	Orthogonal fuzzy neighbourhood discriminant analysis
OLDA	Orthogonal LDA
OS-ELM	Online sequential extreme learning machine
PCA	Principal component analysis
PNN	Probabilistic neural network
Poly-ELM	Polynomial extreme learning machine
PSO	Particle swarm optimization
QDA	Quadrature discriminant analysis
QPC	Quadratic phase coupling
RBF	Restricted Boltzmann machine
RBN	Radial basis network
RegTree	Regression Tree
ReLU	Rectified linear unit
RF	Random forest
RGB	Red Green Blue
RLDA	Regularised linear discriminant analysis
RMS	Root mean square

RVFLN	Random vector functional-link net
SA-EELM	Self-adaptive evolutionally extreme learning machine
SA-SVM	Self-advising SVM
Sig-ELM	Sigmoid extreme learning machine
SFLNs	Single-hidden-layer feed-forward networks
SL	Softmax layer
SOM	Self-organising map
SPCA	Sparse PCA
SR	Spectral regression
SS	Sample skewness
SSC	Slope sign changes
SRDA	Spectral regression discriminant analysis
STFT	Short-time Fourier transform
SVD	Singular value decomposition
SVM	Support vector machine
TD	Time domain
TDD	Time-domain descriptors
TDNN	Time delay neural network
ULDA	Uncorrelated linear discriminant analysis
USELM	Unsupervised extreme learning machine
WAMP	Willison amplitude
W-ELM	Wavelet extreme learning machine
WHO	World health organisation
WL	Waveform length
WPT	Wavelet packet transform
WT	Wavelet transform
ZC	Zero crossing

This page is intentionally left blank

Abstract

A wide gap exists between clinical application results and those from laboratory observations concerning hand rehabilitation devices. In most instances, laboratory observations show superior outcomes the real-time applications demonstrate poor consequences. The robust nature of the electromyography signal and limited laboratory applications are the principal reasons for the gap. This thesis aims to introduce and develop a deep learning model that is capable of learning features from biosignals.

The deep learning model is expected to tame the variable nature of the electromyography signal which will lead to the best available outcomes. Furthermore, the suggested deep learning scheme will be trained to be skilled in learning the best features that match the biosignal application regardless of the number of classes. Moreover, traditional feature extraction is time consuming and extremely reliant on the user's experience and the application. The objective of this research is accomplished via the following four implemented models.

1. Developing a deep learning model via implementing a two-stage autoencoder along with applying different signal representations like spectrogram, wavelet and wavelet packet to tame variations of the electromyography signal. Support vector machine, extreme learning machine with two activation functions (sigmoid and radial basis function) and softmax layer were used for classifications. Moreover, the classifier fusion layer achieved testing accuracy of more than 92% and training attained more than 98%. The same dataset was implemented for superimposed signal representations for two stages autoencoder and softmax layer, support vector machine, k-nearest neighbor and discriminant analysis for classification besides the classifier fusion which led to testing accuracy of more than 90%.
2. Presenting principal component analysis and independent component analysis for feature learning purposes after applying different signal representations algorithms such as spectrogram, wavelet and wavelet packet. Discriminant analysis, extreme learning machine and support vector machine were used for classification. Furthermore, the two proposed models showed acceptable accuracy along

with shorter simulation time. The testing accuracy achieved more than 90% by implementing a classifier fusion layer. Manhattan index was estimated for all features and only the top 50 Manhattan index features were included to decrease the simulation time while attaining acceptable accuracy values.

3. Introducing a self-organising map for deep learning whereby the biosignal was represented by spectrograms, wavelet and wavelet packet. The presented biosignal was introduced to a layer of self-organising map then the suggested system performance was evaluated by extreme learning machine, self-adaptive evolutionally extreme learning machine, discriminant analysis and support vector machine for classification. Adding a classifier fusion layer increased the testing accuracy to 96.60% for ten-finger movements and 99.73% for training. The proposed system showed superior behavior regarding accuracy and simulation time.
4. Presenting a deep learning model where 1) the data was augmented after representing the biosignal by a spectrogram, 2) the augmented signal was represented by a tensor, and finally 3) The signal was introduced to the two-stage autoencoder. The same dataset was used with traditional pattern recognition for comparison purposes. Classifier fusion layer was executed in deep learning scheme whereby the ten-finger movements achieved 90.25% and 87.11% attained by pattern recognition. Besides, the six finger movement dataset was acquired from amputee participants and accomplished 91.85% for deep learning and reached 89.64% for traditional pattern recognition. Furthermore, different datasets for different applications were tested using the recommended deep learning model. Eventually, feeding the deep learning model with various datasets for different applications afforded the model with higher fidelity, combined with real outcomes and generalization.

CHAPTER 1

Problem background

Artificial or prosthetic hands have been considered since ancient times. Prosthetic hands are currently based on identifying the amputee's intent of gesture by analysing an electromyography (EMG) signal. Analysis of electromyography is an established field that has attracted the attention of researchers, but applications of hand rehabilitation in the laboratory show better results than those in actual real life. This was our motivation to "dig deep" to improve the performance of the artificial hand in real life applications by further analysing the electromyography signal. The following two sections will present and explain the problem, followed by our research objectives. After that, we will discuss the thesis findings and the organisation of the thesis. The publications that were produced will then be presented in the last section of this chapter.

1.1 Background-global disability rates

Disability is a human state whereby the word disability is a broad canopy that covers different conditions. An ailment is considered as one condition of disability where it represents the malfunction that may exist in any body function or its structure. A limitation in human activity is another condition which signifies the difficulty that may be met by disabled individuals whereas participation restriction is the barrier that may be encountered by disabled individuals during their participation in life situations.

Everyone is subject to becoming disabled at one stage of his/her entire life, especially older generations. Most families have a disabled member of their family who needs continuous care and support [1-3]. The World Health Organisation (WHO) released a statistic which indicated that 15% of world population live with some disability with 2-4% who encountered substantial functioning difficulties.

In 2011, The World Health Organisation and World Bank's World Disability Report collected disability prevalence data from 59 countries that focuses on functioning-related disabilities. The

report concluded that the occurrence of disability globally was 16% and varied from low-income countries with settled disability occurrence value 18% and 14% for higher income countries. Table 1.1 summaries the disability amount in adults globally.

Table 1. 1: International Disability Occurrence for Adults

Data Source	International occurrence
World Health Surveys[4]	16%
Global Burden of Disease Study[5]	19%

The Global Burden of Disease Study published results based on a different approach to gathering data, which led to 19% as an international disability prevalence [4, 5]. The Washington Group conducted a new study with withdrawing minor and mild disability cases from their studies and considered only substantial disability conditions [6]. Table 1.2 shows the outcome of the Washington Group study.

Table 1. 2: The Washington Group study results [6]

Country	Data Source	Year	Occurrence
Aruba	Census	2010	7%
Israel	Census	2008	6%
Maldives	Survey	2009	10%
United States	Survey	2013	10%
Zambia	Survey	2006	9%

Some studies were concerned about estimating percentages of disability in children. The assessment of disability in children carries higher challenges than in adults as researchers cannot distinguish between the natural growth delay and disability. The results vary widely depending on the methodology that was followed in collecting the data. The United States conducted a study that led to 7% disability percentage for the population who were under 18 years [7]. A more recent study in the United Kingdom led to 7% for the population younger than 18 years [8]. A review conducted in six countries showed a disability percentage range in early childhood of 5% to 12% [9]. These studies showed that the disability occurrence was comparable for both girls and boys

[5]. However, other nationwide studies showed that the prevalence of disability in boys is higher than girls as shown in Figure 1.1.

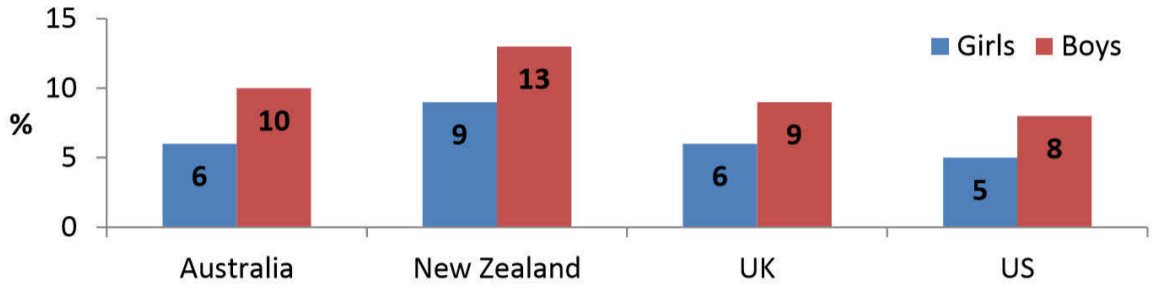


Figure 1. 1: Disability percentage between boys and girls in four countries [10]

Figure 1.1 indicates that in the United States, 8% of boys are disabled whereas 5% is the rate for girls [7]. In the United Kingdom the rate is, 9% for boys and 6% for girls [8]. In New Zealand, boys' percentage was 13% and 9% for girls [10]. Moreover, Australia showed 10% for boys while 6% for girls [11]. Table 1.3 illustrates the disability percentages globally in five countries.

Table 1. 3: The disability percentages for children [8]

Country	Data Source	Age Rate	Disability Percentage
Worldwide	Global Burden of Disease Study	0-14	5%
Australia	Survey of Disability, Ageing and Carers	0-14	8%
New Zealand	National household surveys	0-14	11%
United Kingdom	Blackburn et al	0-18	7%
United States	Newachek and Haflon	0-17	7%

The highest percentage of disability exists in older people. According to a study in Sri Lanka, 65 years or older represented only 7% of the overall population; however, 23% of this age group were disabled [4]. Whereas in Australia, older people occupied 11% from census data, but they account for 23% of total disabilities [4]. The percentage of disability in older people was higher in low-income countries when compared to higher-income countries as shown in Figure 1.2 [12].

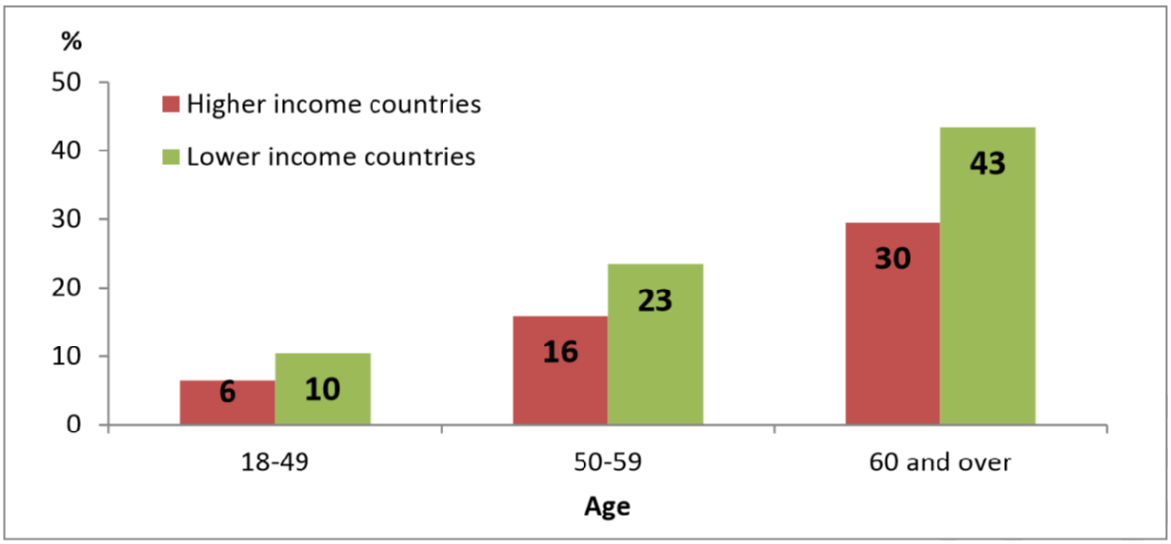


Figure 1. 2: Disability percentage for Low-income and high-income countries [4]

In addition to these studies, Canada conducted a study that showed that the percentage of disabled people was higher in older individuals as presented in Figure 1.3 [13].

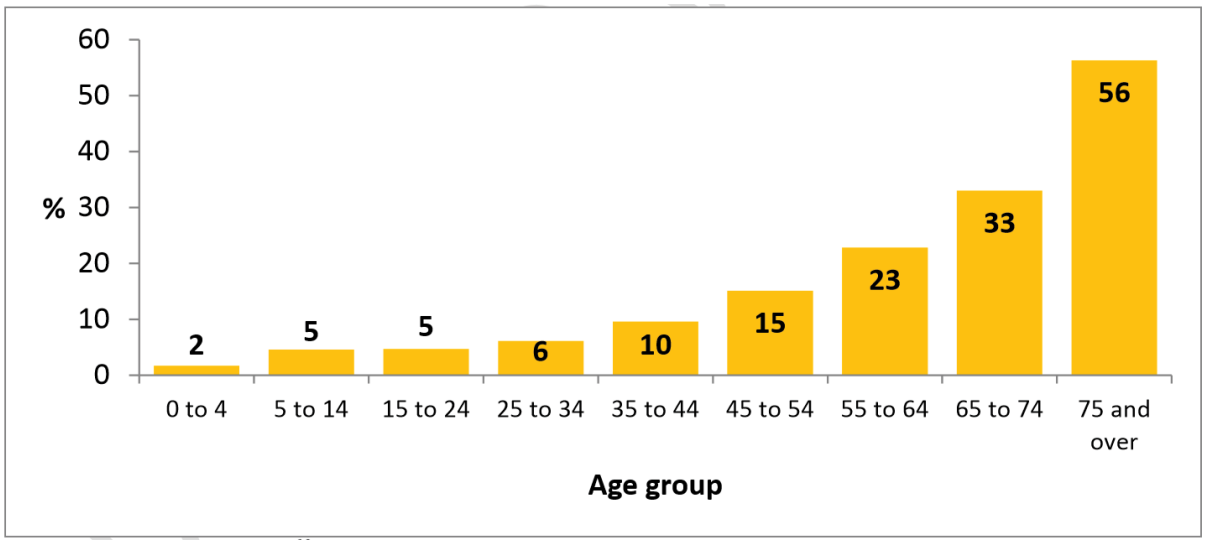


Figure 1. 3: Disability percentage by age (Canadian Study) [13]

1.2 Hand disability

Hand disability is one of the most common disability conditions around the world. Resection or malfunction in hand motors may be the main reason behind disability condition. Regardless of the cause of hand disability, the technology of rehabilitation is a very inspiring mission. The rehabilitation knowledge motivated the researchers to develop many leading-edge technologies.

An innovative artificial hand was presented in 2007 by [14]. Figure 1.4 presents a proposed prosthetic hand.

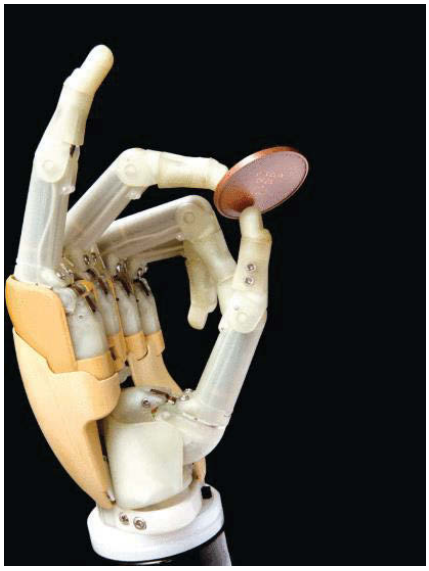


Figure 1. 4: The suggested innovative prosthetic hand (i-limb) [14]

The i-limb was a pioneer to provide the ability for amputee patients to perform movements like a natural hand in clutching objects. Kevin Englehart proposed a control system for upper limb prostheses by using the electromyography signals [15]. Later in 2015, the recommended artificial hand was available for the market [16]. The significant challenge lies in how far the artificial hand will be capable of understanding the patient gesture. The prosthetic hand control system is mainly responsible for simulating the real hand in real-time without any further time delay. Prosthetic hand is not the only technology, but other researchers have introduced rehabilitation techniques [17] whereby the hand was able only to mimic opening and closing and discarded the other gestures. Apparently, this rehabilitation device was for patients who suffered from a failure in their hand functions. The rehabilitation device depended on detecting patient gesture by translating electromyography, which in turn indicated the significance of electromyography pattern recognition. Sensing the gesture of the patient by translating the electromyography signal is a challenging mission. However, it still has many advantages. One of the main advantages is to provide a comfortable and relaxing interface between the patient and the rehabilitation device. This prompted researchers to dig “deeper and deeper” into the field of electromyography pattern

recognition. The brain usually sends signals to muscles to take an action like moving a finger. Researchers targeted the messages that were sent from the brain and received by the muscles, and trained the rehabilitation device to repeat the required action, for example, moving a finger. All the above-mentioned procedures should happen in real time duration without any delay. EMG collection is typically done by placing the electrodes on the muscles by either non-invasive (surface) or invasive (needle) methods. Most patients do not prefer a needle to penetrate their skin and reach the muscle to collect the electromyography signal and would generally prefer non-invasive approaches. Moreover, EMG has a crosstalk that may be caused by the contribution of other muscles. Furthermore, electromyography is characterised by its variations. These drawbacks make collecting and analysing the signal a challenging task. Roche demonstrated a study that aimed to translate the barriers between clinic and laboratory results by collecting EMG and detect the patient gesture of hand intention to move [18]. The muscles that may be responsible for finger movement are not all located in the superficial layers as some are located in the middle and deep layers, which in turn adds difficulty in recording the signal. Understanding the cause of finger movement, either triggered by the firing of one muscle or the contribution of different muscles, is deemed as one of the most common challenges that investigators face in the field of rehabilitation devices.

The limb movements have many actions to perform. Therefore, the rehabilitation device should be capable of sensing the intention of as many movements as possible. The electromyography pattern recognition showed better classification ability than non-pattern recognition, as it was able to trace greater activities by limbs. The electromyography pattern recognition was started by Uchida and others by extracting features from two-channel electromyography signals by implementing fast Fourier Transform (FFT). Uchida had only five-finger movements and applied feed forward neural networks for classification which led to 86% testing accuracy [19]. This accuracy value was accepted at that time. However, the time consumed to build the neural network architecture was the main advantage associated with this approach. Tsenov and his team achieved an enhancement to the Uchida methodology by collecting two-channel electromyography and extracted time domain features where classification was executed by implementing Multilayer perceptron (MLP).

The testing accuracy reached 93% for two-channel and 98% for four-channel electromyography signal. Cipriani et al. collected nine channel electromyography signals from healthy and five amputee participants and extracted time domain (TD) features. They implemented k nearest neighbour (KNN) for classifying seven finger movements. The accuracy was only 79% for amputee participants while it reached 89% for healthy ones. Despite the low accuracy values, the system delay was 50ms for online data. Support vector machine (SVM) is a robust classifier that showed high classification ability. Khushaba and Oskoei used support vector machine as a classifier [20, 21]. Khushaba et al. extracted time domain and autoregressive (AR) features and used linear discriminant analysis (LDA) to reduce features. The results achieved 90% accuracy for online classification and 92% offline. Al-Timemy recorded six-channel EMG signals from six amputee subjects and ten healthy participants to classify twelve to fifteen finger movements [22]. The finger movements were both single and combined actions. He extracted time domain and autoregressive features and reduced the features by utilising orthogonal fuzzy neighbourhood discriminant analysis (OFNDA) [23] with linear discriminant analysis as a classifier. The proposed model achieved 98% accuracy for healthy subjects and 90% for amputees. Despite these promising achievements that were verified in the laboratory, results for real-time applications were relatively low. The main explanation for these differences is that in real applications, the subjects need to perform a large number of movement whereas in the laboratory only a few activities are tested [24]. Moreover, the nature of EMG is variable in the clinical domain [18].

1.3 Summary of challenges

Devices for hand rehabilitation can be divided into two categories: prosthetic or exoskeleton hands. The prosthetic controller is typically implemented by myoelectric pattern recognition. The gap between the accuracy values for amputee subjects and healthy participants is variable according to Al-Timemy [22] whereas the gap is reduced based on Tenore [25]. Hargrove authorised the retrogression between amputee and healthy subjects to the classifier implemented and features extracted. Most of the studies collect EMG signal by using two channel, which provides lower signal details than using a higher number of channels. Khushaba et al. presented a model with a

recorded two-channel EMG signal to classify ten finger movements with 92% as offline accuracy and 89% for online [20]. Aside from robustness and a limited number of finger movements that are usually examined in the laboratory compared to the unlimited number of activities in real-time applications. The EMG signal is also susceptible to the placement and allocation of the electrodes on the skin. Young advised that shifting the electrode position by only 2 cm caused a deterioration in the performance by 25-30% [26]. Further, the setup of the experiment and surrounding circumstances could affect the results, which in turn bring the concerns about how far myoelectric pattern recognition can be adapted with changes that may occur to the EMG.

In contrast, exoskeleton hands are very primitive as they typically can close or open only. Wege implemented a model from ten electromyography channel data that should be able to introduce more movements to the exoskeleton hand but unfortunately, this model did not show any successful performance [27].

1.4 Objectives

As was mentioned in the previous sections, there are many problems surrounding collection and analysis of electromyography signals, however, we believe that reliable signal analysis will lead to a convenient and reliable rehabilitation device that should be capable of satisfying patients' needs. In this research, we are interested in the development and enhancement of signal representation, features extraction and feature learning. We believe that signal representation and best features extracted or learnt will lead to high accuracy values.

The objectives of this research are listed as follows:

a. Introduce powerful signal representation techniques.
b. Investigate feature extraction and feature learning systems, and apply them to biomedical data.
c. Develop a novel deep learning system that should be capable of learning features by self-learning, and apply different classification techniques to best classify ten finger movements from only two-channel EMG.

d. Develop a deep learning approach that should be able to classify the signal independent of the application for either finger movements or other biomedical data.

e. Develop an augmentation technique along with tensor representation before introducing the biomedical data to the deep layer of feature learning and optimising our classification methodologies.

f. Develop a reliable pattern recognition system to deliver high performance for both amputees and non-amputees.

1.5 Contribution to the doctoral thesis

This thesis is interested in implementing deep learning, which incorporates serveral feature learning concepts. The main objective of this dissertation is to develop a deep learning model with a robust or variable signal representation technique and powerful classification layer. It will combine a middle-deep neural network layer that is trained on learning features by itself without any human intervention in selecting the features that best match the application. To achieve the recommended model we verified different enhancements in the field of feature learning.

- The first attempt to develop a self-learning model with a robust signal representation technique was utilises independent component analysis (ICA) as a feature learning technique with applying different signal representations algorithms (spectrogram, wavelet and wavelet packet). The performance of the suggested scheme will be evaluated by applying different classifiers such as Support Vector Machine, Extreme Learning Machine and Discriminant Analysis. Moreover, to improve these results, a classifier fusion layer will be implemented to select the most accurate result for both training and testing datasets. Classifier fusion layer resulted in a promising training and testing accuracies. Data from our laboratory showed that the training accuracy was more than 98% whereas testing was more than 86%. Moreover, the sequential feature selection technique was applied to the same dataset. The selected features were further reduced, and the model performance was evaluated by holdout and cross-validation. The cross-validation missclassification error for a limited number of features was less than 1%. The data were recorded using two-

channel EMG to classify ten finger movements. Further details and explanation for these results will be mentioned in Chapter 5. This paper was presented at 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA) 2016 and published in IEEE under title “ICA based feature learning and feature selection”.

- The second attempt was introducing principal component analysis (PCA) as a feature learning layer for the same data set. The biosignal was robustly represented and well prepared before being introduced to feature learning layer. Support vector machine, extreme learning machine and discriminant analysis were used as classifiers and classifier fusion technique was used to enhance the proposed model performance. The number of features was reduced by estimating the highest Manhattan index. PCA showed better performance than ICA as training accuracy was more than 98% and testing achieved values higher than 90% for classifying ten finger movements. More details and explanation for these findings will be mentioned in Chapter 5. This paper was presented at 8th Cairo International Biomedical Engineering Conference (CIBEC), 2016 and published in IEEE under the title “PCA indexing based feature learning and feature selection”.
- The third attempt was implementing a self-organising map neural network as a feature learning layer. The biosignal was well prepared to be presented to the feature learning layer. Evolutionally extreme learning machine was well explained and analysed and utilised as a classifier technique besides traditional extreme learning machine, support vector machine and discriminant analysis. The confidence interval for these mentioned classifiers was estimated with a confidence score of 60% and analysis of variance (ANOVA) was evaluated. The implementation of a classifier fusion layer promoted the accuracy values for training accuracy to higher than 99.5%, and for testing, to 96.6%. This journal paper will be explained and demonstrated in details in chapter 6, and it was published in Advances in Science, Technology and Engineering Systems Journal (ASTESJ) with title “Self-Organizing Map based Feature Learning in Bio-Signal Processing”.

- Two stage of autoencoder results were utilised for feature learning. The same efficient signal representation algorithms were used in this paper. Softmax layer, extreme learning machine and support vector machine were used as classifiers. In this model, different activation functions for extreme learning machine were presented as sigmoid, radial basis function. Moreover, the rectified linear unit (ReLU) was designed as an activation function. The ReLU with activation function showed better classification ability when compared to the sigmoid approach. Both ANOVA and confidence interval was valued for all classifiers with a confidence score of 60%. Classifier fusion layer brought promising results for our proposed model as the training accuracy was higher than 98.5% and testing was greater than 92%. Further details and clarifications will be explained in Chapter 4. This paper was submitted in Advances in Science, Technology and Engineering Systems Journal (ASTESJ). In addition to the above-mentioned approach, the same configuration of the autoencoder was recalled while the biosignal representation with the combination of spectrogram, wavelet and wavelet packet approaches. Softmax, k-nearest neighbour, discriminant analysis and support vector machine were applied for classification. Classifier fusion layer was used based on best local classifier methodology. The window size values were inspected to select the best value that led to the highest accuracy values. The training accuracy was above 98%, and testing achieved almost 91% with a window size of 200msec. Further explanation will be discussed in Chapter 4. This paper was presented in The First MoHESR and HCED Iraqi Scholars Conference in Australasia 2017 with title "Superimposed Signal Representation for Deep Learning in Biosignal Processing".
- Given the focus on signal representation, we suggested a scheme that should be applicable to any bio signal regardless of the application. To our knowledge, with rich data in terms of size and number of channels, the accuracy value is likely to be higher. Based on this information, a data augmentation technique was introduced to our spectrogram represented raw biosignal. The data augmentation followed Gaussian theory. After that, the augmented data were fed to a tensor layer to add more degrees of freedom to our data. Then the data were presented to the two-layer autoencoder, and softmax layer, extreme learning machine

and support vector machine were applied for classification. Furthermore, classifier fusion was executed to develop the results. This model was applied on different biosignal applications similar to the collected data for ten-finger movements. The same pattern with the same configuration was applied on Iris, Breast Cancer, Seeds, Sonar, Mines vs. Rocks and Indian Liver Patient Dataset. Moreover, the recommended system was executed on data collected from nine amputee participants with eight channel EMG. The model using different data sets achieved accuracy of 98.5% for testing. The main advantage of this model was that it could be considered reliable and any biosignal application could be executed. More descriptions and details will be clarified in Chapter 7.

1.6 Organization of the thesis

This thesis contains eight chapters and references. The chapters are organised as follows:

i. Chapter 1 Introduction
ii. Chapter 2 Literature review
iii. Chapter 3 Methods for feature learning, deep learning techniques and signal representation methods
iv. Chapter 4 Autoencoder model implementation for deep learning
v. Chapter 5 Implementation of PCA and ICA for feature learning
vi. Chapter 6 The utilisation of self-organising neural networks for deep learning
vii. Chapter 7 Powerful signal representation techniques combined with deep neural networks
viii. Chapter 8 Thesis summary, conclusion and recommendations for future work

1.7 Publications outcome of the doctoral research

The publications during my doctoral research are as follows:

Submitted Journal

- **Marwa Farouk Ibrahim Ibrahim**, Mohammad Omar Wedyan, Adel Ali Al-Jumaily "Bio signal Data Augmentation and Tensor Representation for Surface Electromyography Deep Learning", Biomedical Signal Processing and Control, ELSEVIER Journal.

Published Journal

- **Marwa Farouk Ibrahim Ibrahim**, Adel Ali Al-Jumaily "Self-Organizing Map based Feature Learning in Bio-Signal Processing", Advances in Science, Technology and Engineering Systems Journal (ASTESJ), Volume 2, Issue 3, Page No 505-512.
- **Marwa Farouk Ibrahim Ibrahim**, Adel Ali Al-Jumaily "Auto-Encoder based Deep Learning for Surface Electromyography Signal Processing", Advances in Science, Technology and Engineering Systems Journal (ASTESJ).

Published Conferences

- **Marwa Farouk Ibrahim Ibrahim**, Adel Ali Al-Jumaily (6-8 Dec. 2016). "ICA based feature learning and feature selection". Paper presented at 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA), Ras Al Khaimah, United Arab Emirates and published at IEEE.
- **Marwa Farouk Ibrahim Ibrahim**, Adel Ali Al-Jumaily (15-17 Dec. 2016). "PCA indexing based feature learning and feature selection". Paper presented at 8th Cairo International Biomedical Engineering Conference (CIBEC), Cairo, Egypt and published at IEEE.
- **Marwa Farouk Ibrahim Ibrahim**, Sahar Adil Abboud, Adel Ali Al-Jumaily (5-6 Dec. 2017). "Superimposed Signal Representation for Deep Learning in Biosignal Processing". Paper presented at The First MoHESR and HCED Iraqi Scholars Conference in Australasia, Melbourne, Australia.

CHAPTER 2

Literature review

This chapter demonstrates a review on hand anatomy and recognizes the assembly of the hand that any rehabilitation device should be adaptable to it. Moreover, biosignals' characteristics will be illustrated while focusing on electromyography one. Furthermore, the previous work and contributions that have been done in the field of finger movement will be presented. Later, an overview of feature selection techniques, methods, and give a brief on favorite features and its types like time domain, frequency domain will be explained. Finally, the results of different feature selection algorithm with different classifiers will be compared.

2.1 Hand Anatomy

In this section, the hand dissection will be demonstrated in detail. Hand anatomy is required, in this study, as every motion is resulted from the cooperation of each hand element. The hand anatomy is very sophisticated and is needed for the proper design of rehabilitation devices. Some scholars consider human hand to consist of four fingers and thumb [28] while others scholars consider it to consist of five fingers [29]. The main elements of the hand are: bones, Skin, nails, palmar fascia and deep compartments, nerves, muscles and tendons, joints, blood Supply and pulley.

2.1.1 Bones

The hand contains 27 bones excluding the sesamoid bone [30]. The hand bones have three sets. The first is the set of carpals, the second is the set of metacarpals, and the third is the phalanges. Figure 2.1 shows the groups of bones in the hand. Hand has eight carpal bones and five metacarpals [31]. The bone of the wrist is believed to be the most complicated joint in the human body. The wrist has eight carpal bones, which are divided into two sets with limited motion between those two groups. Figure 2.2 displays a dorsal view of the hand bones while Figure 2.3 presents a palmar view.

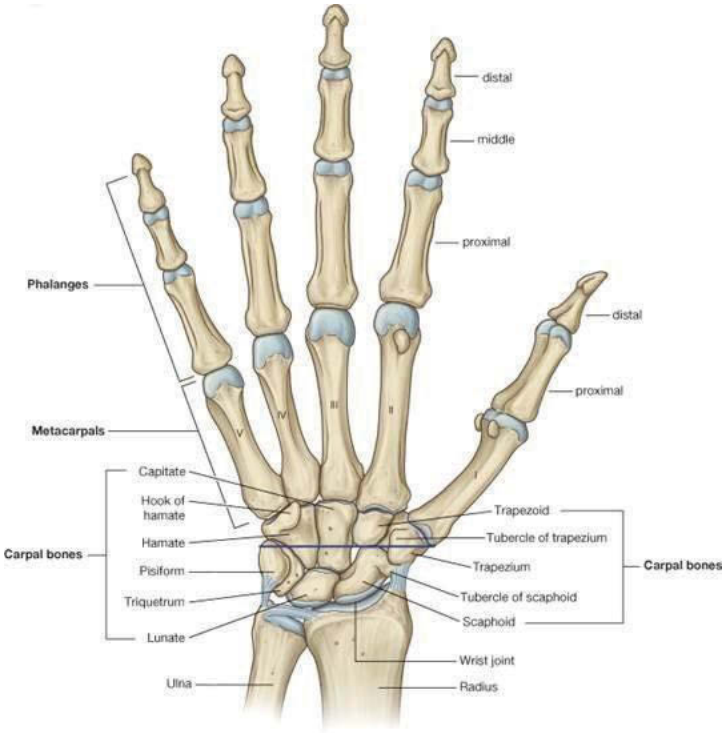


Figure 2. 1: Bones groups in hand [31]

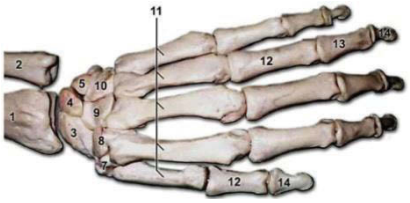


Figure 2. 2: Hand Bones dorsal view [31]

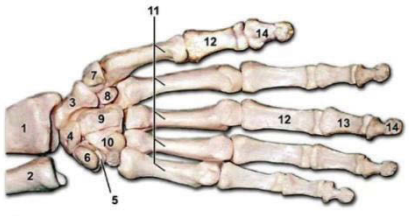


Figure 2. 3: Hand Bones Palmar view [31]

In Figure 2.2 and 2.3 Bone 1 represents the radius, Bone 2 is the ulna, Bone 3 denotes the scaphoid, Bone 4 designates the lunate. Whereas bone 5 indicates the triquetral, bone 6 signifies the pisiform, bone 7 specifies the trapezium, bone 8 assigns the trapezoid. In addition, bone 9 shows the capitate, bone 10 demonstrates the hamate, bone 11 illustrates the metacarpal, bone 12 clarifies proximal phalanx, bone 13 displays middle phalanx and bone 14 indicates the distal phalanx. The two sets of wrist bones are estimated from radial to ulnar. The proximal row involves the scaphoid, lunate, triquetrum, and pisiform bones whereas the distal row includes the trapezium, trapezoid, capitate, and hamate bones. The Pisiform bone is the only bone that not does contribute to wrist movement whereas all other carpal bones join in the contribution to any function for the wrist. The pisiform is defined as the sesamoid bone that the flexor carpi ulnaris tendon passes through. The scaphoid acts as the connection between the two groups of wrist bones, which in turn make it frequently subject to breakage. The distal row of wrist bones is firmly joined to the base of the second and third metacarpals. This stable unit acts as a reference to the other moving units. The flexor retinaculum makes the surface of the carpal tunnel. The flexor retinaculum is joined to the trapezium radially, the scaphoid, the hook of hamate ulnarly and the pisiform. Every finger has one metacarpal bone, so each hand has five metacarpal bones. Each metacarpal bone has a head, a neck, a base, and a shaft. The thumb metacarpal bone is the most movable and shortest bone [32]. The thumb metacarpal bone articulates with the trapezium while the four metacarpal bones articulate with the hamate, the capitate and the trapezoid at the base. The head of Metacarpal bone for each finger is muscularly joined to with the proximal phalanges. The hand encloses 14 phalanges. Each finger has three phalanges (proximal, middle, and distal) whereas the thumb has only two phalanges. All of the bones mentioned above form the radicle of the hand and its fingers [33].

2.1.2 Skin

The skin is the external layer that covers the surface of the hands. However, the skin that covers the palm is entirely different from the skin that covers the dorsum as shown in Figure 2.4. In Figure 2.4, Number 1 represents the extensor retinaculum, Number 2

denotes the flexor retinaculum, Number 3 indicates the head of the metacarpals, Number 4 designates the ulnar artery, Number 5 signifies the radial artery, Number 6 specifies the depth of palmar arch and Number 7 illustrates the deepth of superficial palmar arch.

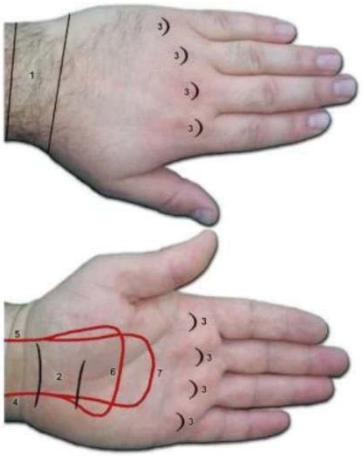


Figure 2. 4: Hand skin for palm and dorsum [31]

The hand palmar skin is exceptional as it has many functions. The palmar skin is thick, bald and not as flexible as dorsal hand skin. The hand palmar skin is connected to the fascia via several vertical fibres. The characteristics of hand palm skin allow for a proper grip.

2.1.3 Nerves

Three nerves support the hand; the median, ulnar, and radial nerve. Each of these nerves consists of sensor and motor components. The hand nerves are different from the traditional nerves [34]. The forearm skin is supported, in a median manner, by the medial antebrachial cutaneous nerve and horizontally by the lateral antebrachial cutaneous nerve.

2.1.3A Median Nerve

The median nerve is responsible for controlling the muscles used in the gripping functions of the hand. The median nerve is located in the horizontal and medial cords

of the brachial plexus (C5-T1). The branches of the motor control the flexor digitorum superficialis, flexor carpi radialis, pronator teres and palmaris longus muscles in the forearm. The anterior interosseous branch controls the pronator quadratus, flexor pollicis longus, and flexor digitorum profundus muscles, which move the index and long fingers. A sensation at the thenar eminence is supplied by the palmar cutaneous branch which is the closest to the wrist.

The thenar muscles (superficial head of flexor pollicis brevis, abductor pollicis brevis, and opponens pollicis) are controlled by the recurrent motor branch as the median nerve goes along the carpal tunnel. Sensation is granted by digital sensory branches to the long, index, thumb, and radial side of the ring finger.

2.1.3B Ulnar Nerve

The ulnar support the muscles that are responsible for strong clenching of the hand. It is located at the horizontal cord of the brachial plexus (C8-T1). The motor branches of ulnar nerve controls both the flexor digitorum profundus and the flexor carpi ulnaris muscles of the small and ring fingers. Sensation at the hypothenar eminence is provided by the palmar cutaneous branch. The ulnar nerve grants sensation, located at the dorsum of the hand, to part of the ring finger and small finger via the dorsal branch. The dorsal branch is a branch from the main trunk positioned the distant forearm. Sensation is provided to both ulnar of the ring and small fingers by the digital nerves.

Both the ulnar artery and the deep motor branch go along the Guyon canal. The hand deep motor is essential and has the main function as its controls the hypothenar muscles.

2.1.3C Radial Nerve

The radial nerve controls the extenders of the wrist which in turn positions and stabilizes the hand. The radial nerve is located at the cord of the brachial plexus (C6-8). The radial motor branch controls both the extensor carpi radialis longus and brachioradialis muscles at the elbow. As it is the closest nerve to the forearm, the radial

nerve is categorised into deep and superficial branches. The deep branch controls all the muscles located in the extender section.

2.1.4 Muscles and Tendons

Hand muscles are classified into intrinsic and extrinsic classes. The intrinsic muscles are substantial ones and are positioned in the hand while the extrinsic ones are existing nearby the forearm and introduced to the skeleton of the hand through long tendons [35].

2.1.4A Extrinsic extensors

All of the extensor muscles are believed to be extrinsic excluding the interosseous-lumbrical complex. The interosseous-lumbrical complex is engaged in interphalangeal joint extension. The radial nerve supports the extrinsic extensor muscles entirely. The extrinsic extensor muscles contain a significant combination of digit and thumb extensors as well as three wrist extensors [36]. The extensor carpi radialis longus (ECRL) moves the wrist in a radial fashion whereas the extensor carpi ulnaris (ECU) moves in an ulnar one. The extensor carpi radialis brevis (ECRB) along with the extensor carpi ulnaris and the extensor carpi radialis longus are representing the prime wrist extensors. The ECRB is positioned in the third metacarpal, the ECRL is located in the second metacarpal, and the ECU is situated in the fifth metacarpal.

The extensor digiti minimi, extensor indicis proprius, and extensor digitorum communis expand the fingers. Those three extensors are located at the bases of the medium phalanges as principal sliding bonds and the distal phalanges as side bonds. The extensor pollicis longus, abductor pollicis longus and extensor pollicis brevis extend the thumb. The thumb extensors are attached to the basis of the thumb metacarpal, aftermost and distant phalanx. The extensor retinaculum prohibits the tendons from strangling at the wrist and detaches them into six units. The extensor digitorum communis is a sequence of tendons attached to each finger through a shared muscle belly, along with inter tendinous bridges among them. The extensors for the

small and index fingers are both independent. The index extensor is the extensor indicis proprius, whereas the small finger extensor is the extensor digiti minimi.

2.1.4B Extrinsic flexors

The extrinsic flexors have three wrist flexors along with a considerable combination of thumb and finger flexors. The median nerve controls all these flexors excluding the flexor digitorum profundus and the flexor carpi ulnaris (FCU). These are attached to the ring, and ulnar nerve supports the small finger.

The flexor carpi ulnaris, flexor carpi radialis and palmaris longus form the significant flexors in the wrist although the palmaris longus is not present in 15% of individuals [37]. The palmaris longus is attached to the palmar fascia whereas the flexor carpi radialis is connected to the base of the third metacarpal and the flexor carpi ulnaris is linked to the base of the fifth metacarpal. The flexors are classed as either surface or profound. The flexors and the flexor pollicis longus move across the carpal tunnel to flex at the interphalangeal joints. The flexor pollicis longus is connected to the distant thumb phalanx. The surface tendons of the flexor digitorum are located volar to the profundus tendon. The superficial tendons divide close to the phalanx and bond away from the profundus tendon to attach to the middle phalanx.

2.1.4C Intrinsics

The intrinsic muscles are located entirely within the hand and are categorised into four sets: the interossei, lumbrical, hypothenar and thenar muscles.

The interossei set has three palmar muscles and four abaxial ones, all controlled by the ulnar nerve. They start at the metacarpals and shape the horizontal bonds with the lumbricals. The dorsal interossei supports the fingers whereas the abaxial interossei ranges from the fingers to the palm centreline.

The lumbrical set contributes to the extension of the interphalangeal joints and flexion of the metacarpophalangeal joints. They emanate from the flexor digitorum profundus tendons of the hand and attach to the radial side of the extensor tendons of the fingers.

The median nerve supports the long finger and index lumbricals whereas the ulnar nerve supplies the ring and small finger lumbricals [38].

The hypothenar muscles comprise of the opponens digiti minimi, abductor digiti minimi, flexor digiti minimi and palmaris brevis, which are controlled by the ulnar nerve. This set of muscles are connected to the carpal bones and the flexor retinaculum and are inseted in the base of the small finger proximal phalanx.

The thenar set has the adductor pollicis, flexor pollicis brevis, opponens pollicis and abductor pollicis brevis muscles. The median nerve controls the abductor pollicis brevis and the opponens pollicis whereas the ulnar nerve controls the adductor pollicis and the flexor pollicis brevis.

2.1.5 Joints

The joint of the wrist is very complicated as it is multi-hinge one. It permits a wide range of movements. The distant radioulnar joint permits palm pronation and supination as the radius turns around the ulna. The radiocarpal joint comprises of the distant radius and the proximal carpal bones. Both the ulna and the radius articulates with the proximal row of carpals to control both ulnar and radial deviation along with flexion and extensions. The radius and ulna produce an external set of powerful volar bonds to control this joint. The dorsal radiocarpal and intercarpal bonds support the joint.

The movement between carpal bones is confined at the intercarpal joints. Robust inherent bonds reinforce the intercarpal joints. The lunotriquetral and the scapholunate bonds are the most significant ones at the intercarpal joints. Any interruption or break that may happen to either of these bonds will lead to unreliability in the wrist. The distal carpal bones and the proximal carpal bones shape the smaller and larger arc respectively. The metacarpals are forming joints with the four distal carpal bones at the carpometacarpal (CMC) Joints. The headmost CMC joint composes the motion unit while both second and third CMC joints shape the stationary one. The collateral bonds

restrict the horizontal movement. This configuration and organisation of the bonds grants bonds close fitting during joint tightness and free ones during joint expansion. The palmar plate provides the hyperextension. The palmar plate is one of the parts that composes the joint capsule and fastens to the proximal phalanx. The palmar plate, at the interphalangeal joints, restricts the expansion. The palmar plate fastens to both surfaces of the joint. The collateral bonds limit the ulnar and radial movement as it becomes taut during those gestures.

2.2 Biosignals

A Biosignal is believed to be any signal that can be recorded from living organisms. The expression biosignal can be called on both electrical and non-electrical signal. The biosignal is characterised by being stochastic and time-varying. The electrical biosignal is defined by the potential difference that is produced across any living organism tissue, organ or cell. The most popular biosignal are as follow:

Electromyogram (EMG), Electroencephalogram (EEG), Electrocardiogram (ECG), Electrooculography (EOG), Galvanic skin response (GSR), Magnetoencephalogram (MEG) and Mechanomyogram (MMG).

EMGs, EEGs, ECGs and EOGs are measured across the potential difference between two poles of electrodes using either an invasive or non-invasive method. GSR or Electrodermal activity (EDA) measures the electrical resistance of the skin in response to any variation. Whereas MEG measures the magnetic field that is produced by brain natural electrical activity. A collection of biosignals may be collected by invasive or non-invasive methods, nevertheless other remote technologies were introduced and utilised [39-43]. In this thesis, the electromyography surface signal will be further analysed and applied therefore a useful background on EMG signal will be presented.

2.2A Electromyography

Electromyography is the biosignal that is collected and recorded from the electrical activity of the skeletal muscles [44]. The electromyography signal is generated when the cell of the muscle is activated or stimulated which in turn causes muscle contraction and potential differences that can be detected by surface electrodes or invasive needles [45]. So in other words, the electromyography signal emanates from the change that may occur in the membrane of the muscle fibre [46]. In another word, the EMG is the sum of excitation signals from many muscle fibres. The signal is sometimes called a myoelectric signal (MES) or an electromyogram (EMG) [47]. Figure 2.5 shows the EMG signal pattern.

Figure 2. 5: EMG signal pattern [48]

It is clear from figure 2.5 that the electromyography signal is of a stochastic or random nature. The level of the signal varies from zero to 1.5mv as a root mean square value or from zero to 10mv (-5 mv to +5 mv) as a peak to peak. The electromyography resembles any other electrical signal regarding its susceptibility to noise. The main frequency power range is located from 20 to 150 Hz whereas the electromyography signal ranges from 6 to 500 Hz [49]. Figure 2.6 exhibits the EMG signal pattern and its power.

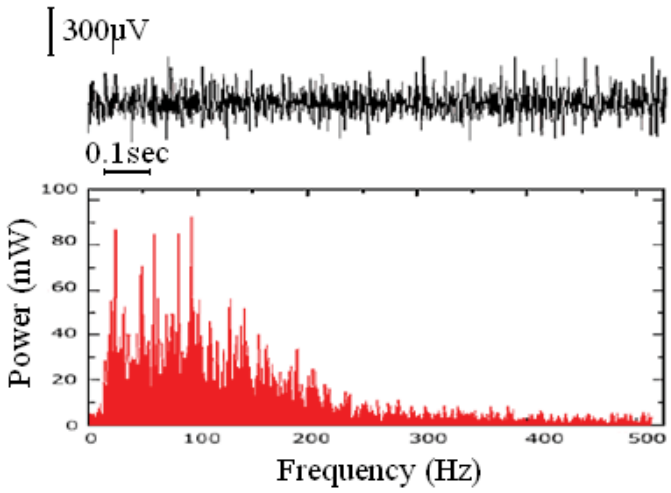


Figure 2. 6: EMG signal pattern in black and its energy in red [49]

The motor unit represents the body of the cell with its branched extension of a nerve cell (dendrites); it includes several muscle branches with the innervating muscle fibre. All the muscle fibres behave as one unit while supplying any organ with nerves or innervation procedure, that is why the expression unit is called motor unit [50]. In another word, a motor unit is a group of muscle fibres innervated by a single motor neuron. The number of motor units per muscle may vary widely from hundreds to thousands depending on the muscle as the muscle may be a small hand muscle or a large limb one [51]. Figure 2.7 represents the motor unit.

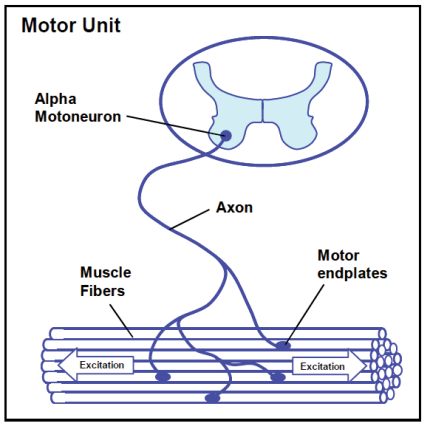


Figure 2. 7: Motor unit [48]

The muscle has a resting potential that is generally from -80 or -90 mv. This value represents the muscle under rest: no contraction or tension. The potential difference describes the voltage difference between the intracell and the outer surface. The resting potential difference may increase by applying any activation or stimulation to the muscle through a depolarisation procedure. However, activation may not occur when stimulating the muscle membrane with a value less than the threshold. The muscle will be activated for a duration of time and then return to its resting potential by a repolarisation process. A semi-permeable membrane as shown in Figure 2.8 can model the resting potential, depolarisation, and repolarisation activities.

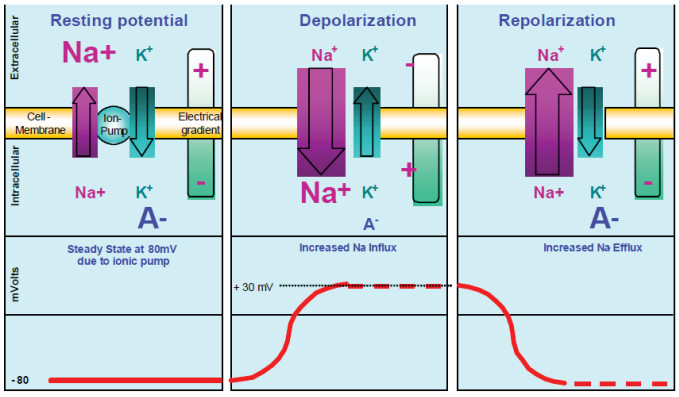


Figure 2. 8: Depolarisation and polarisation of muscle membrane excitation [48]

The Na^+ inflow increases until reaching a certain threshold value at which the depolarisation starts to occur. The depolarisation results in a potential change from -80 or -90 mv to +30 mv for a specific duration which is known as action potential. Then a repolarisation process will restore the rest potential to establish after a hyperpolarisation interval. Figure 2.9 clarifies the action potential method.

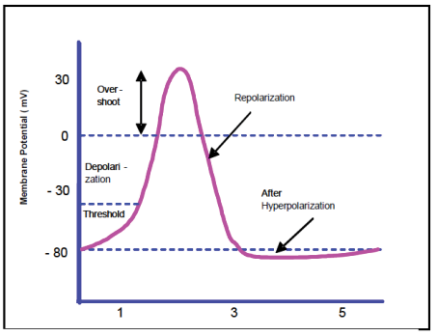


Figure 2. 9: The Action potential [48]

The studies show that the action potential for all motor units is the superposition of each motor unit individually, which generates a symmetric electromyography signal around the zero. Therefore, the amplitude of the signal in the positive section is identical to that in the negative one. Figure 2.10 displays the summation of each motor unit action potential.

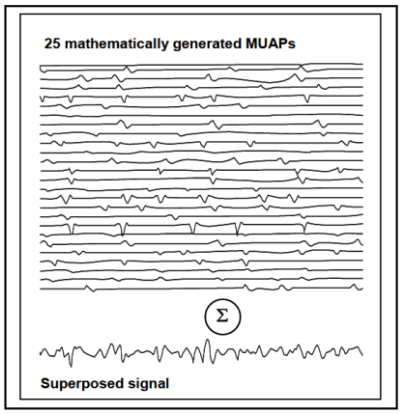


Figure 2. 10: The superimposition of motor unit action potential [48]

Both the firing frequency and the recruitment of the motor unit control the density and the magnitude of the resultant electromyography signal. The human skin behaves as a filter, so the recorded signal is not equal to the original one, but it acts as a representative of it. Figure 2.11 exhibits the firing frequency and recruitment of multiple motor unit.

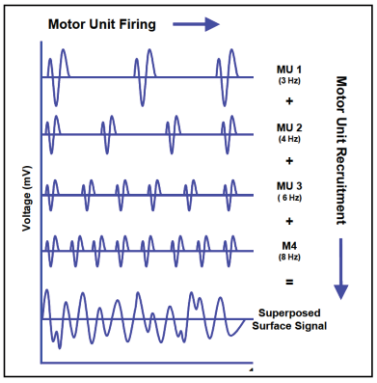


Figure 2. 11: The superimposition of multiple motor unit with different firing frequencies [48]

The electromyography can be represented as shown in figure 2.12.

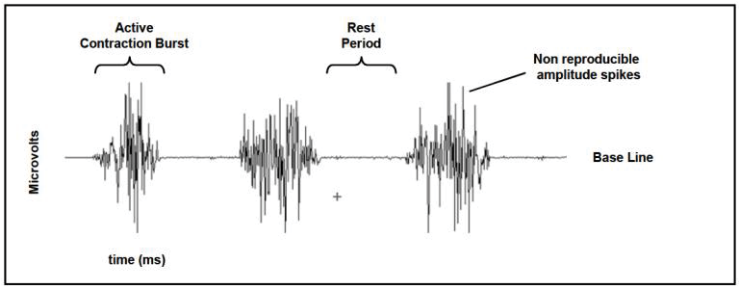


Figure 2. 12: The raw EMG for three contractions recorded from the biceps brachii muscle [48]

As shown in figure 2.12, three contractions perform the EMG signal of the biceps brachii muscle. The signal amplitude is in millivolts and is symmetric around the baseline. The signal has rest periods representing the nonactivational duration of the muscles (where the muscles were in rest), whereas the three contraction bursts describe the activation of the muscles. Each burst has spikes during the activation of the muscle. The baseline represents the muscle during relaxation where no contraction exists. The baseline can be clean or noisy, this mainly depends on the quality of the collecting electrodes, the quality and reliability of the used amplifiers and the environment if it is noisy or not.

The average noise level should vary between 3 to 5 microvolts when hiring an innovative amplifier with a proper skin cleaning and preparation while utilising high-quality electrodes.

2.2A.A Electromyography skin preparation

The placement of non-polarizable electrodes, proper skin cleaning, and preparation are the main aspects that may lead to reliable EMG signal. The cutting-edge technology of EMG amplifiers released amplifiers, which were prepared for skin impedance between electrodes range from 5 to 50k ohm. Removing any extra hair is an essential step in preparing the skin as it helps in adhering the electrodes, particularly in a humid environment or for perspiring skin. Moreover, cleaning the skin is an essential move towards acquiring a reliable, accurate EMG signal. Many methods could be applied to lead to a clean skin. The first method is using paste to remove any dead skin, accumulated dirt, or moisture on the skin. The second favorite method is to apply sandpaper with gentle rubbing, as harsh rubbing will cause skin irritation. Alcohol cleaning pads should be used alongside the sandpaper. The third method is employing pure alcohol along with a cotton towel and rubbing gently to avoid skin inflammation. As mentioned above, EMG recording may be accomplished via invasive or non-invasive (using surface electrodes) methods [52]. Surface methods are more comfortable to the patient as no needle or wire penetrates the patient's skin. The only drawback of the surface electrodes is that they can only be used for surface muscles and not deeper ones. Two types of electrodes are currently sold on the market. The top one is a wet gel electrode which has a thin layer of gel added. The wet gel type is recommended as it guarantees no sanitary issues that may arise from implementing these disposable electrodes. Additionally, it provides a low impedance contact. The other type is an adhesive gel electrode, which is superior only in case of allocating electrode in the wrong position as it can be repositioned. However, both kinds are disposable as they are strictly for one person only.

2.2A.B Electromyography signal detection

There is no practical difference between the control system for invasive and non-invasive signal collection methods. Figure 2.13 describes how the EMG signal is generated from motor units that in turn contribute to superimpose each motor unit segment to produce the last electromyography signal.

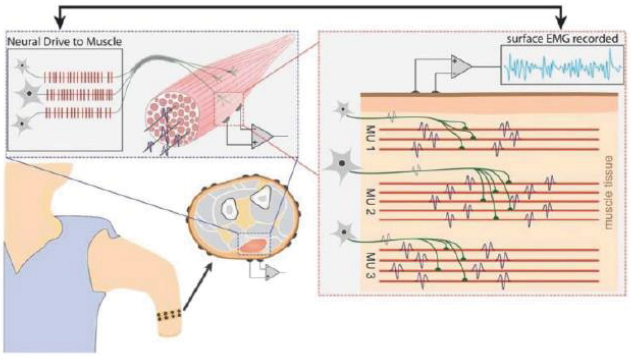


Figure 2. 13: The production and recording of EMG [18]

Figure 2.14 illustrates the procedures for EMG data acquisition. The motor unit is acting as the source of any EMG signal where the action potential is activated through muscle contraction. The measuring electrodes read the superimposed signal for all activities [53].

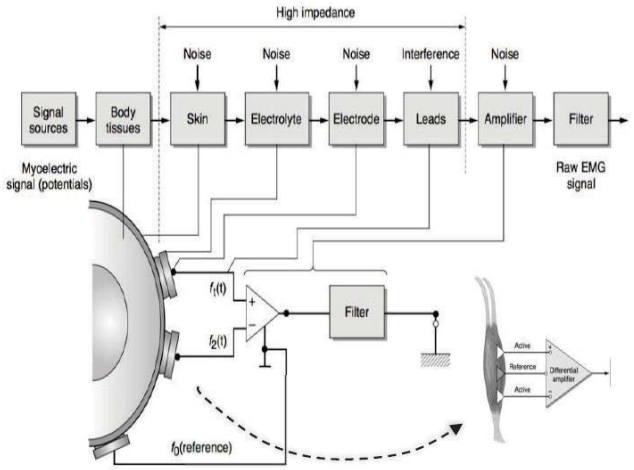


Figure 2. 14: EMG Recording stages and tools [53]

As it is clear from figure 2.14 that the noise may be added to the signal in more than one stage. We aim to collect EMG signal without noise, or with minimal noise and mostly information. The noise could be acquired from the skin, electrodes, cables, amplifiers and the surrounding environment. Applying differential amplifier clue, with high common mode rejection ratio to suppress the noise, amplify the difference signal and expel any common one, which in turn reduce the noise level accumulated on the original signal. Furthermore, concerning on the design of high-quality electronic devices, including the power cables, will have an important impact on reducing the level of noise into a minimum.

2.2A.C Factors affecting EMG signal

Many factors may affect the recorded EMG signal, like the characteristics of the subject tissue. The human body by itself is conductive, however; its conductivity is profoundly affected by the tissue kind, its thickness and temperature. Crosstalk from neighbouring muscles may cause interference with the targeted signal. As for example, collecting the electromyography signal of the shoulder's muscles attracts the EGG signal to intervene with the electromyography signal spikes precisely as shown in figure 2.15. As it is clear from figure 2.15, we have a consistent EMG signal with a massive interfering EGG signal in the middle of the band, and with a comparable magnitude to that for the EMG signal. Moreover, any movement in the electrode allocation will have its adverse impact on the detected signal. Noise from either the skin, electrodes, or amplifier (including all the electrical cables) should be reduced to a minimum to acquire a reliable signal.

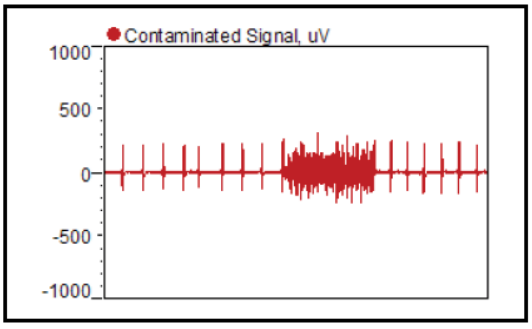


Figure 2. 15: EGG interference with EMG [48]

2.2A.D Benefits of EMG signal

The EMG has many benefits and advantages in different fields. In addition to the medical importance of the electromyography signal, it may also be beneficial in the field of rehabilitation, research, sports, and any dealings between humans and their surrounding working or industrial environment. The EMG signal gives us the opportunity to read the muscle performance, analyse and evaluate it. It helps physicians make decisions before or post-surgery. In addition, it prevents risk either in sports or in working environments. Figure 2.16 explains the different applications areas for electromyography.

Figure 2. 16: EMG different applications areas

2.3 Finger Movement Classification

Many scholars and researchers suggested models that were capable of classifying finger movements by using biosignals or other methodologies. The purpose of precisely classifying finger movements was to predict the intention of the finger to movement and translate hand gestures accurately. This proper understanding and translation give the opportunity to develop more advanced rehabilitation devices. These state of the art rehabilitation devices will in return allow a higher rate of healing for patients. The proposed finger movements and hand gesture classification models will be presented in this section.

Researchers used Electrocorticographic (ECOG) signal to classify five finger movements [54]. The authors recruited six subjects to collect ECOG signals from, and classified five-finger movements: thumb, fore, mid, ring, and little. The authors used an 8x8 platinum electrode array

with only a 1 cm gap between each electrode. Each finger movement was repeated for only two seconds and with break intervals. The Electrocorticographic signal is collected from the brain and is usually used to determine the patients with intractable epilepsy. The concept of using ECOG rather than Electroencephalographic (EEG) was based on the results that were concluded from [55] which stated that the ECOG signal contained more information in the higher frequency band than those available in the EEG. Moreover, the power stored in the high-frequency band showed a distinguished stimulation pattern for different sections of the body [56]. The authors used the same features that had been implemented before in a previous study conducted on tongue movements [57]. Also, the scholars concluded that the ECOG signal could be classified more precisely than EEG while using lower training data. The authors divided the collected signal into three bands: low, mid, and high-frequency bands. The low band range was 11–40Hz, whereas the mid band was 71–100Hz and the high band range was 101–150Hz. Another study paid attention to the visibility of implementing the high-frequency band spectral power features [58]. The investigators utilised two feature selection strategies. The first one was targeting the three power bands and deduced that, the mid-power and high-power band features led to better and distinct features than the low band. Whereas the second strategy was the active feature selection strategy suggested in [59]. The concept of the latter strategy was ranking the features based on their discriminative ability and choosing only the highest ones. The authors implemented two linear classifiers; a support vector machine (SVM) and a linear programming machine (LPM).

The researchers assumed that they had data of x_k with equivalent classes or labels y_k where $y_k \in \{+1, -1\}$. The labels were allocated the label $sign\ (w^T x_k + b)$. Where the parameters (w, b) were estimated from the data by applying optimisation techniques to decrease the possibility of misclassification to its lowest values. They applied the following criterion

$$min_{w,\xi,b} \frac{1}{2} ||w||_2^2 + \frac{C}{K} ||\xi||_1 . \tag{2.1}$$

Whereas the labels or the classes are estimated from the following formula

$$y_k\,(w^T + x_k + b) \geq 1 - \xi_k, \quad \text{and}$$

(2.2)

$$\xi_k \geq 0 \quad for\ k = 1, \ldots . K(1)\ .$$

(2.3)

$||.||_1$ signifies the l_1 norm (in another words $||w||_1 = \sum |\,\omega_i|$) whereas $||.||_2$ denotes the l_2 norm or the Euclidean. The term ξ_k , slack variable, indicates the trade-off between the misclassification error and the superiority of the classifier. Also, the parameter C has a role balancing this compromise and is selected empirically using cross validation. The term $\frac{1}{2}\ ||w||_2^2$ introduces a margin.

The LPM was derived from the SVM by substituting l_1 norm for l_2 [60]. LPM introduces sparse weight vectors where most of its components are zero or close to zero, that in turn related any non-zero values to the feature selection algorithm and made the classifier more robust to noise. Where the LPM could be expressed as follow

$$min_{w,\xi,b}\ \frac{1}{N}\|w\|_1 + \frac{C}{K}\ ||\xi||_1\ .$$

(2.4)

Whereas the labels or the classes are estimated from the following formula

$$y_k\,(w^T + x_k + b) \geq 1 - \xi_k, \quad \text{and}$$

(2.5)

$$\xi_k \geq 0 \quad for\ k = 1, \ldots . K\ .$$

(2.6)

Figure 2.17 shows the percentage of error that resulted from implementing the suggested model using LPM and SVM classifiers. It is clear that, LPM showed better classification ability than the SVM due to sparsity term. However. The overall behaviour needed improvement to obtain higher accuracy values.

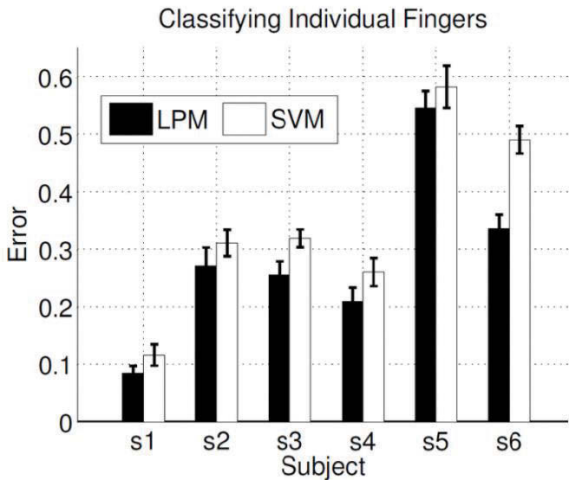


Figure 2. 17: Error percentage for classifying five-finger movements for six subjects [61]

Another research recruited the electromyography signal to classify three finger movements via fuzzy systems [61]. The signal was recorded from the surface of the forearm by hiring only four electrodes as shown in figure 2.18.

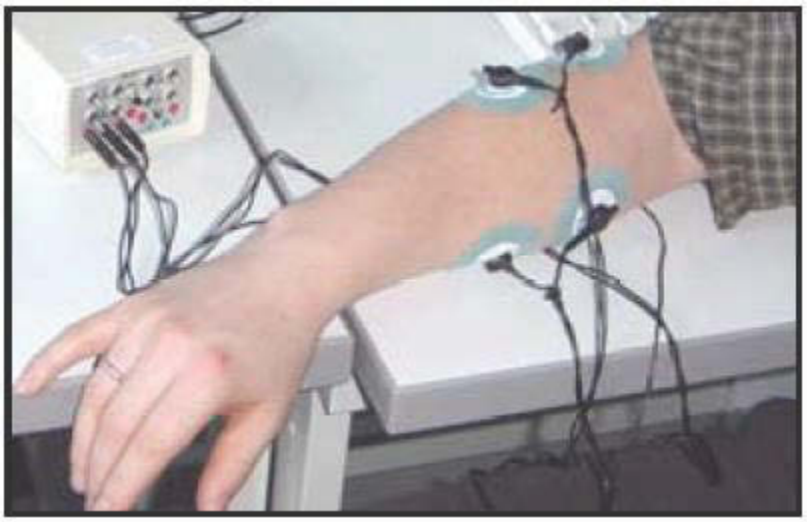


Figure 2. 18: EMG signal recording via four electrodes [61]

The collection of EMG signals from an amputee is very complicated due to the loss of many muscles [62]. Extensor digitorum and Palnaris longus were the two muscles sets that the signal was recorded from. The three targeted classes were thumb, pointer and middle. The scholars used transient signals rather than the steady state signals as the

transients had more information than the steady state signal. Moreover, they calculated short time Fourier transform (STFT) as a time-frequency analysis to recognise the frequency content of the signal as well as its development across the time. The authors applied a gain of 2000 and sampled the signal at 4000 Hz. The academics utilised the power spectral density (PSD) as a feature extraction technique. Figure 2.19 shows the normalised distribution for both training and testing sets.

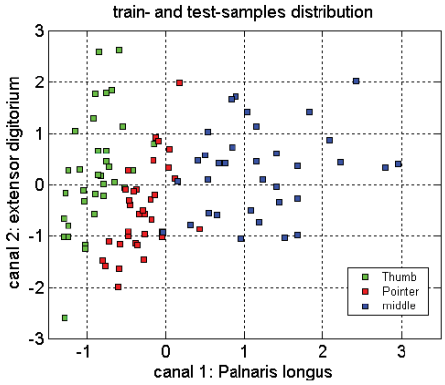


Figure 2. 19: Normalised training and testing distribution [61]

The authors recruited a technique for the initialisation of fuzzy sets called ellipsoidal delimitation. The initialisation basis was obtained from the available input and output data set. The input data were defined as X_{nf} whereas the output was described as Y_n. Where n was the number of samples and $f$represented the number of features. $i = 1, \dots, n$ and $f = 1, \dots, f$. The authors used the trimmed mean method [63] and reduced the similar fuzzy sets into one set to avoid redundancy. The mean for both features and classes were estimated as follow:

For features $F_j = X_{ij}\ (i = 1, \dots, n_k)$ While for classes $C_k, k = 1, \dots, K$

$$V_{jk} = \frac{1}{n} \sum_{i=1}^{n_k} X_{ij} \ . \tag{2.7}$$

n_k Identifies the number of samples equivalent to class C_k with K classes. The total number of samples for the whole classes could be described as $n = \sum_{k=1}^{K} n_k$.

Figure 2.20 displays the ellipsoidal delimitation distribution for trimmed training and testing set.

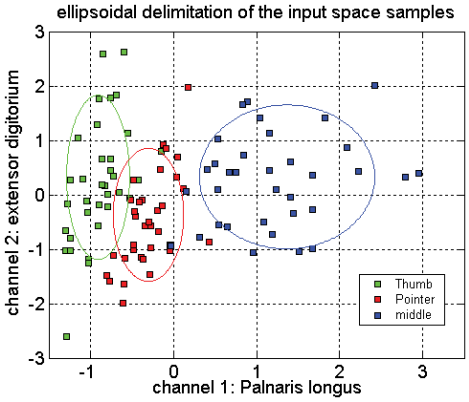


Figure 2. 20: ellipsoidal delimitation distribution for trimmed training and testing set [61]

The accuracy values reached 86.27% whereas after removing redundant fuzzy sets with optimisation the accuracy value became 92.16%.

In the same context of finger movements' classification, the researchers in [22] conducted an in-depth study on ten sound-limbed subjects and six amputee participants. The study recruited 12 surface EMG channels for intact subjects and 11 for amputee participants as shown in Figure 2.21.

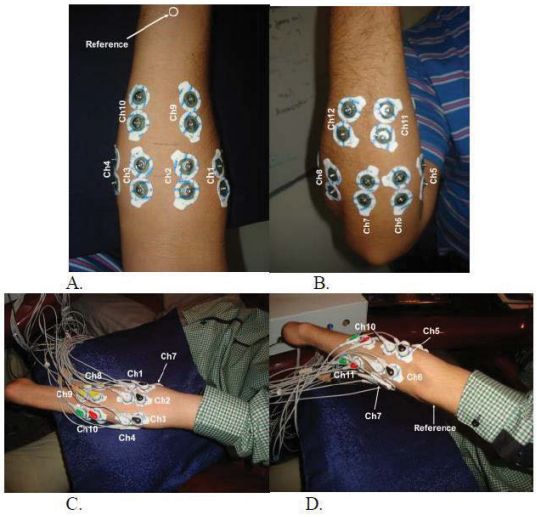


Figure 2. 21: A. Front view for electrodes allocation on intact subjects [22]

B. Rear view for electrodes allocation on intact subjects [22]

C. Front view for electrodes allocation on amputee subjects [22]

D. Rear view for electrodes allocation on amputee subjects [22]

The sound-limbed subjects were asked to perform 15 and 12 classes whereas the amputee participants were asked to do the 12 classes only. The utilised electrodes were self-adhesive Ag–AgCl. The authors of this research used many electrodes although lowering the number of electrodes had its impact on decreasing the difficulty, time and cost for the executed controller to resolve [52]. The European endorsements to allocate electrodes positions and the 24 mm distance between electrodes were followed in this research [26, 64]. The scholars used time domain autoregressive (TD-AR) feature extraction technique due to its benefits. TD-AR allowed for actual time enactment and added robustness to the signal against any movement of electrodes [15, 65]. The academics employed principal component analysis (PCA) and orthogonal fuzzy Neighborhood discriminant analysis (OFNDA) for feature reduction whereas SVM and Linear discriminant analysis (LDA) for classification. They recruited different combinations of implemented feature reduction and classifications techniques as shown in Figure 2.22.

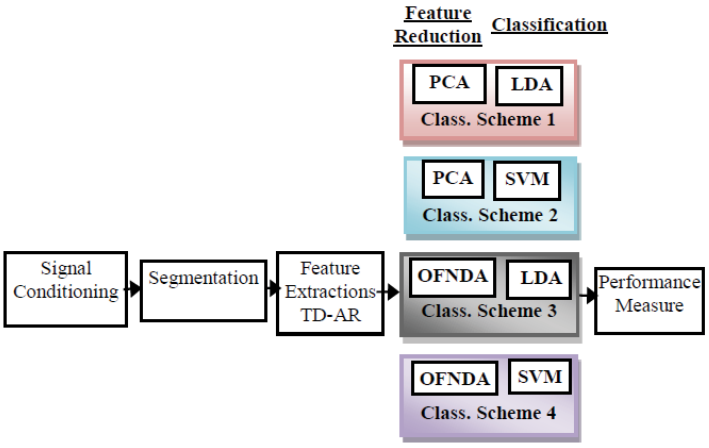


Figure 2. 22: Suggested classification system with four combined schemes [22]

The proposed model led to the results shown in Figure 2.23 with a classification error versus a number of channels as displayed in Figure 2.24.

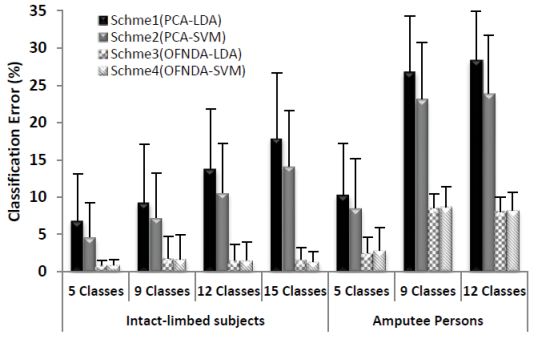


Figure 2. 23: Implementation of error for number of classes with four schemes [22]

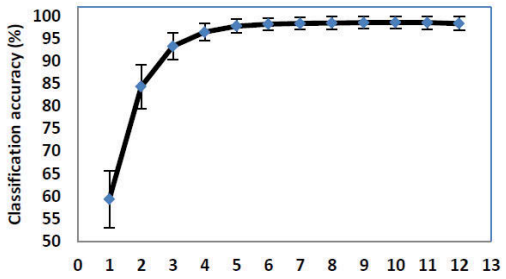


Figure 2. 24: Classification error for proposed model versus number of channels [22]

Necmettin Sezgin [66] conducted a study that recruited two-channel surface electrodes to collect electromyography signal and categorise hand finger gestures. The signal was recorded from extensor and flexors muscles that were located in the forearm. The author applied bicoherence analysis to the collected signals and used extreme learning machine (ELM) as a classifier. The bicoherence of the signal is the normalised release of the bispectrum where the bispectrum is the Fourier transform for the third order cumulative $C_{3x}(n_1,n_2)$. The bispectrum is a statistical methodology that can disclose any nonlinear component that exists in the signals [67]. The bispectrum was implemented and led to substantial results especially in analysing nonstationary signals like EEG, EMG and ECG [68-71]. The bispectrum showed better performance than power spectral density with non-Gaussian and nonlinear signal as it provided more information [72]. The proposed model is shown in Figure 2.25.

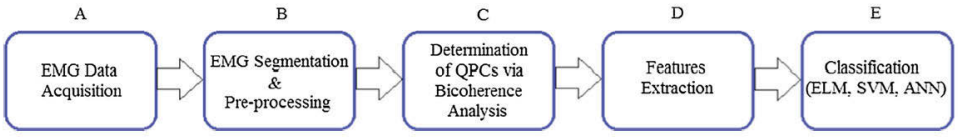


Figure 2. 25: The suggested scheme block diagram [66]

Assume that we have an EMG signal $x(k)$. The third order accumulative can be estimated via as follows [73]:

$$C_{3x}(n_1,n_2) = Cum\{x(k), x(k+n_1), x(k+n_2)\}. \tag{2.8}$$

Or

$$C_{3x}(n_1,n_2) = E\{x(k)x(k+n_1)x(k+n_2)\}. \tag{2.9}$$

Where $E\{\cdot\}$ designates the anticipated values. The third order cumulative can be rephrased as follows:

$$C_{3x}(n_1,n_2) = m_{3x}(n_1,n_2) - m_{3x}^{G}(n_1,n_2).$$

(2.10)

Where $m_{3x}(n_1, n_2)$ identifies the third order moment of the input EMG signal $x(k)$ whereas $m_{3x}^{G}(n_1, n_2)$ denotes the third order Gaussian random process for the same signal $x(k)$ and the third order cumulative with zero mean can be expressed as follows:

$$C_{3x}(n_1, n_2) = m_{3x}(n_1, n_2) .$$

(2.11)

As the bispectrum is the third order transform and based on the equations mentioned above, the bispectrum $B(\omega_1, \omega_2)$ can be defined as follows:

$$B(\omega_1, \omega_2) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} C_{3x}(n_1, n_2) W(n_1, n_2) e^{-j(\omega_1 n_1 + \omega_2 n_2)} .$$

(2.12)

The $W(n_1, n_2)$ represents the window function where this study recruited hanning window with 0.1 seconds as a window duration. The bispectrum can be defined as Fourier transform as follows:

$$B(\omega_1, \omega_2) = < X(\omega_1) X(\omega_2) X^*(\omega_1 + \omega_2) > ,$$

(2.13)

Where * represents the complex conjugate. Once the bicoherence is the normalised version of the bispectrum so it can be expressed as follows:

$$b^2(\omega_1, \omega_2) = \frac{|B(\omega_1, \omega_2)|^2}{|X(\omega_1, \omega_2)|^2 |X(\omega_1) X(\omega_1)|^2} .$$

(2.14)

The author collected biosignals from 42 participants: 20 females and 22 males. The data were collected for five trials and each trial duration was 10 seconds. The recorded signal was sampled at frequency 1000 Hz and amplified by gain of 2000. The scholar implemented a sixth order Butterworth filter with cutoff frequency band from 2 Hz to 500 Hz to remove any noise that might be superimposed on the original EMG signal. The bicoherence was estimated, and the quadratic phase coupling (QPC) was attained

as well. Moreover, the entropy, mean, variance and root mean square for the Bicoherence was considered as features. The researcher aimed to classify five finger movements by using only two surface electrodes and with applying 10-fold cross-validation for the signal. Figures 2.26 and 2.27 show the accuracy versus the number of neurons by applying the ELM classifier for all subjects and each subject respectively.

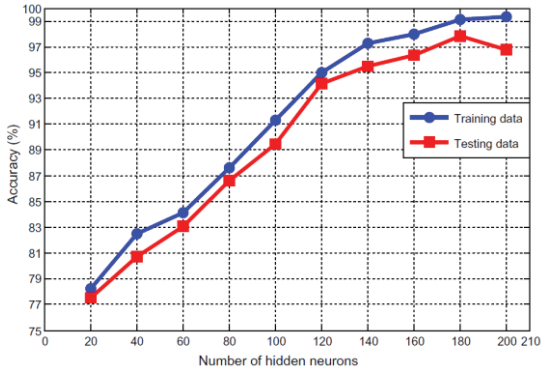


Figure 2. 26: Accuracy versus number of neurons for all participants [66]

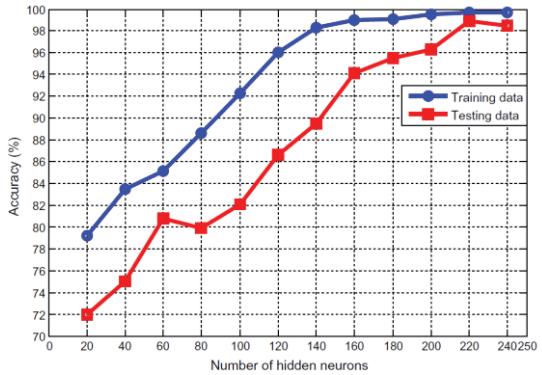


Figure 2. 27: Accuracy versus number of neurons subject based [66]

Anam et al. conducted different studies in the field of finger classification as per [74]. Anam et al. utilised two surface electromyography channels to record single and combined finger movements. The author used spectral regression discriminant Analysis (SRDA) for feature reduction, ELM for classification purposes and majority voting for smoothing the classification. SRDA showed faster performance than LDA. Moreover, ELM performed faster than SVM while achieved the same classification

rendering as widely used SVM classifier. The proposed model achieved 98% accuracy for both single and combined finger movements. Anam et al. applied the same suggested model for online and offline data and verified accuracy 97.96 % and 97.07% respectively [75] for classifying ten finger movements. The researcher recorded two-surface electromyography from five amputee participants. Anam et al. optimised the ELM classifier by implementing particle swarm optimisation (PSO). The primary purpose of this suggested model was to classify 11 finger gestures. The results were compared with both the standard PSO and the grid search methodology. The recommended model reached 94% accuracy for five amputee participants [76]. Anam et al. published another study to classify between only two gestures from the index finger [77]. The first gesture was flexion (F) while the other one was at rest (R). Anam et al. collected the surface electromyography signal by recruiting only one channel from ten healthy participants (four females and six males). The signal was iterated six times for each gesture with five seconds as rest between each trial. A gain of 1000 and 2000 MHz sampling frequency were applied. Figure 2.28 displays a diagram of the proposed model.

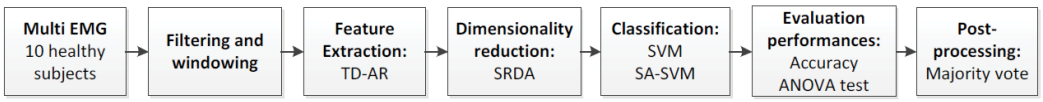


Figure 2. 28: Block Diagram for the suggested pattern recognition model [77]

Time domain and autoregressive features were extracted from the one-channel electromyography signal after being filtered by the bandpass filter of bandwidth from 20 Hz to 500 Hz. The extracted time domain features were sample skewness (SS), slope Sign changes (SSC), waveform length (WL) and zero crossings (ZC). Moreover, the Auto-regressive (AR) and Hjorth time domain parameters (HTD) were extracted as well. AR was sustainable to any shift in the electrode positioning along with rapid variations in the signal levels [78]. Furthermore, SRDA was fast and achieved a good performance for a single electrode and a large number of data. Anam et al.

implemented self-advising SVM (SA-SVM) for the classification and it showed 0.63% in the accuracy levels. SA-SVM generated advice weights from the training data to be used in the test data set to remove any misclassification that exists in the testing stage, which in turn increased the accuracy levels. Four-fold cross-validation was executed. The analysis of variance (ANOVA) was recalled as another estimation of the accuracy besides the traditional classification techniques. Majority voting was used to improve the classifier. SVM and SA-SVM achieved average accuracy values 99.0107 % and 99.6411% respectively. In a study relating to finger gestures recognition, Anam et al. collected the electromyography signals from two-channels only for eight subjects to classify ten finger movements [79]. The researcher implemented the same model shown in Figure 2.28 except for classification of ELM with three different activation functions. Anam et al. applied adaptive wavelet extreme learning machine (AW-ELM), wavelet extreme learning machine (W-ELM) and sigmoid extreme learning machine (Sig-ELM) as activation functions for the ELM classifier. Figure 2.29 shows the executed model.

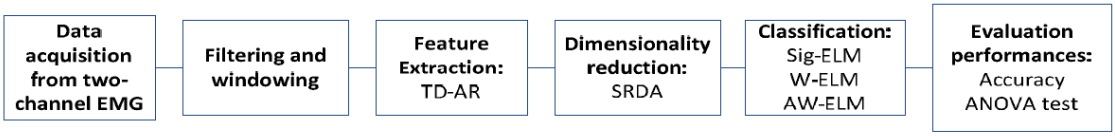


Figure 2. 29: Block Diagram for classifying ten finger movements by using AW-ELM [79]

The data were filtered and windowed by applying 200 ms as a window duration with 25 ms as an increment. 200 ms was chosen to simulate the real-time applications. Additionally, the same set of feature extraction and reduction methodologies that had been applied before were used in this proposal. The only difference was in the classification technique as in this research proposal; the utilised classifier was ELM with three different activation functions. AW-ELM showed the best performance among the other two implemented activation functions. The accuracy exceeded 91%

for the three activation functions. However, AW-ELM was superior for the most number of nodes except for 50 and 75, where Sig-ELM showed better performance while W-ELM was the worst performance among the three. In [80] the author suggested a new feature reduction technique where he combined ELM with spectral regression (SR). The weights for implemented ELM were estimated randomly while those for the output were valued using spectral regression. The integrated scheme was called SRELM. This study recruited eight subjects and collected surface EMG data from two muscles in the forearm. The first muscle was the flexor digitorium superificialis whereas the second was the flexor policis longus. The author made a comparison between different feature reduction algorithms and the suggested one, SRELM. The executed feature reduction techniques were SRELM, SRDA, uncorrelated linear discriminant analysis (ULDA), PCA, OFNDA, unsupervised extreme learning machine (USELM) and without any feature reduction (baseline). Figure 2.30 shows the average accuracy percentage for each feature reduction algorithm versus the number of classes.

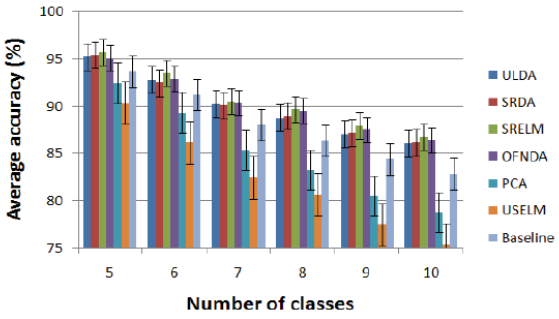


Figure 2. 30: Suggested model performance for different feature reduction algorithms versus the number of classes [80]

The values of the number of hidden nodes and the regression coefficient α both controlled the system performance. Figure 2.31 demonstrates the accuracy value

against the number of hidden nodes whereas Figure 2.32 expresses the accuracy value versus the regression coefficient.

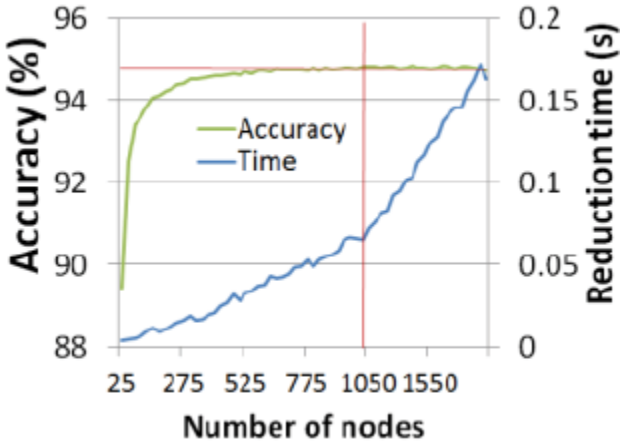


Figure 2. 31: Suggested model accuracy value versus number of nodes [80]

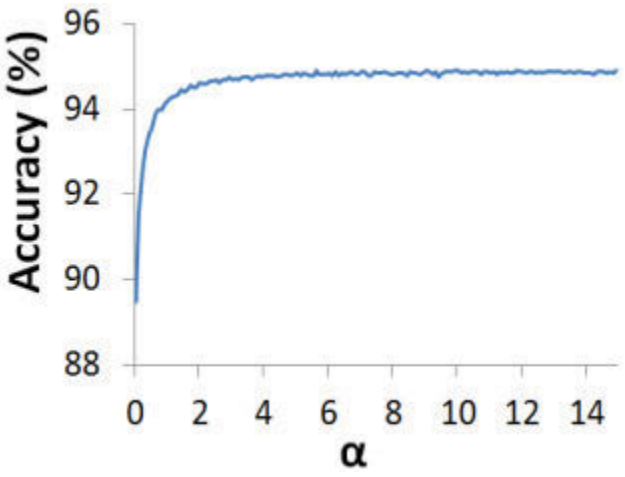


Figure 2. 32: Suggested model accuracy value versus regression coefficient α [80]

In [81], Anam et al. collected surface electromyography signals from eight healthy participants. The author implemented three stages of classification. The first was offline classification, the second was adaptation stage and the third was online classification. The subjects were asked to perform ten finger gestures. Five of them

were single-finger movements whereas the other five were combined movements (two fingers at a time or hand close). The same set of time domain features that had been extracted before was extracted beside the Hjorth and autoregressive parameters. Spectral regression discriminant analysis (SRDA) was used for feature reduction whereas online sequential extreme learning (OS-ELM) was applied for classification purposes. OS-ELM divides the training data into segments and learns it segment-by-segment. Classification via OS-ELM has two phases. The first phase was initialisation and the second phase was sequential. The training data were divided into an initialisation set and a sequential set where the data were processed segement-by-segment. The initialisation set had a concise period of data 0.5 s and the sequential set has the rest of data with 100 ms as an increment. The author applied three-fold cross-validation on the data set. The second phase was the adaptation stage where data collected for a short duration of time. The collection of data for the adaptation stage acquired new settings including the electrodes positioning which in return required new training and adjustment from OS-ELM. The third phase was the online classification in this stage; the participants were asked to perform free actions and specific gestures by their hand. The resultant signal, in both stages, was analysed by the above-mentioned technique. The author implemented different classifiers beside OS-ELM to evaluate their performance in comparison with other classifiers such as SVM, radial basis function ELM (RBF-ELM) and AW-ELM. Figure 2.33 reveals the behaviour of each classifier per subject while the overall performance is written at the bottom of the figure.

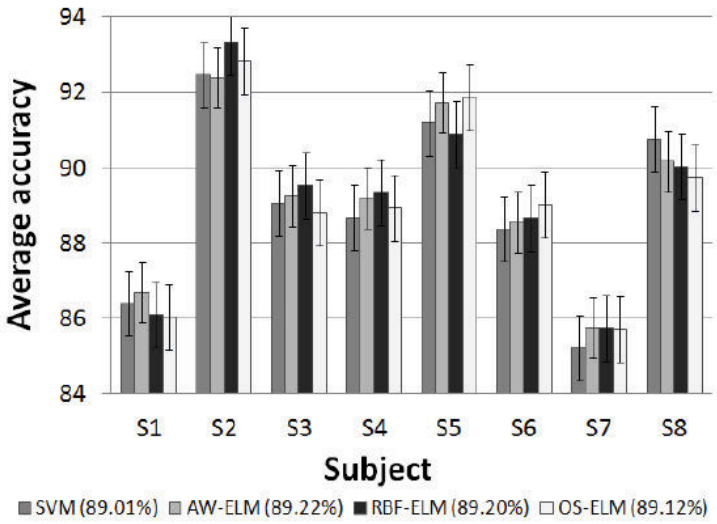


Figure 2. 33: Average accuracy for all subjects and individually for four classifiers [81]

In addition, the researcher varied the number of hidden nodes during the offline classification to pick up the optimum number, which was 110 nodes based on Figure 2.34. The 110 hidden nodes were used for the online classification.

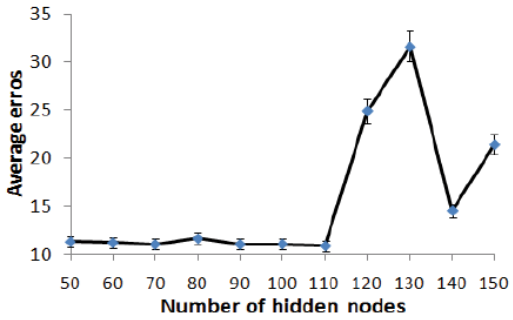


Figure 2. 34: Average accuracy against the number of hidden nodes during offline classification [81]

The OS-ELM showed a robust ability to adapt itself to be in line with the new settings as shown in Figure 2.35. Where the OS-ELM showed better stability and even higher

accuracy than RBF-ELM for data that were collected one or two days ago while RBF presented a degradation in the accuracy levels for the second and third day.

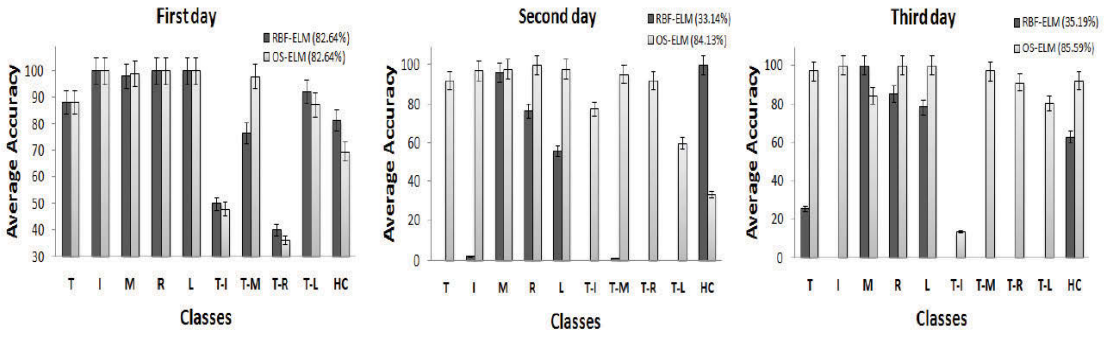


Figure 2. 35: Average accuracy for subject 1 for three consecutive days [81]

Recently, Anam et al. published a new study that led to classifying finger movements with accuracy exceeding 99% [82]. The author processed surface electromyography signals that were collected by Al-Timemy [22] where twelve surface channels were positioned on the right forearm for nine healthy participants and eleven surface channels were positioned on the forearm for five amputee subjects who suffered from shock beneath the elbow. Figure 2.36 shows the electrodes allocation for both healthy and amputee participants.

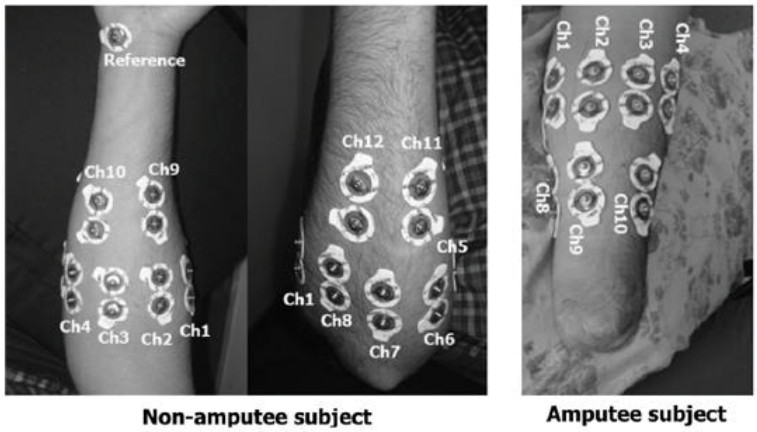


Figure 2. 36: Electrodes allocation for healthy and amputee participants [22]

The EMG data acquisition device had an amplification of 1000 that was applied to the acquired signal. Moreover, it had two analogue filters. The first one was a low pass filter with cutoff frequency 450 Hz whereas the second filter was a high pass filter with cutoff frequency 10 Hz. The low pass filter was fourth order Butterworth whereas the high pass was second order Butterworth. The sampling frequency was 2000 Hz. Furthermore, the acquisition device had two extra digital filters where the first one was a bandpass filter (20-450 Hz) whereas the second was a notch filter at 50 Hz from the fifth order Butterworth. Healthy subjects were asked to perform eleven finger movements, three combined gestures and one rest. Amputee participants were requested to accomplish eleven gestures with one rest. Figure 2.37 displays the proposed system.

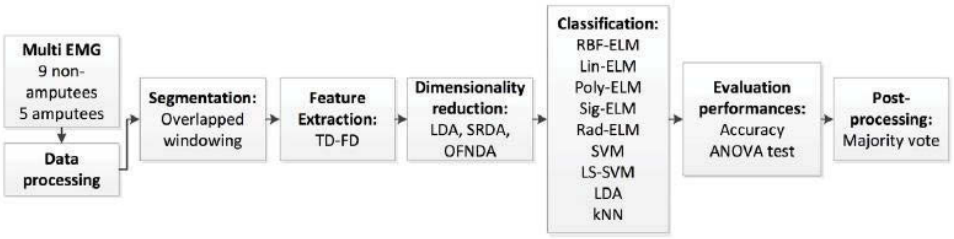


Figure 2. 37: Block diagram for the suggested system [82]

It is clear from Figure 2.37 that the researcher used window segmentation after processing the data where the window length was estimated via equation (2.15) to avoid using window length that causes delay.

$$D = \frac{1}{2}T_{wl} + \frac{n}{2}T_{inc} + \tau \ , \tag{2.15}$$

Where D is the delay time, T_{wl} is the window length, T_{inc} is the window increment and τ is the processing time that was consumed by the implemented pattern recognition system. Then some features were extracted from the signal whereas here the author combined time domain and autoregressive features to extract the most useful information from the signal. Thereafter, the dimension was reduced by applying LDA,

SRDA and OFNDA. Subsequently, the researcher applied various classifiers to carry out the classification task. Anam et al. employed the ELM classifier with different activation functions such as sigmoid additive hidden-node (Sig-ELM), multi-quadratic RBF hidden node (Rad-ELM), RBF kernels (RBF-ELM), polynomial (Poly-ELM) and linear (Lin-ELM). Beside the ELM different classifiers, the author implemented LDA, SVM, least-square SVM (LS-SVM) and k-nearest neighbour (kNN) as famous classifiers which would help in evaluating the performance of different ELM classifiers in comparison with the widely used classifiers. As per other publications of Anam et al., he calculated ANOVA as another aspect to evaluate system performance and used majority vote to smooth the classifier performance. Figure 2.38 shows the accuracy percentage versus number of channels while Figure 2.39 displays the accuracy against window length and Figure 2.40 shows the processing time versus window length.

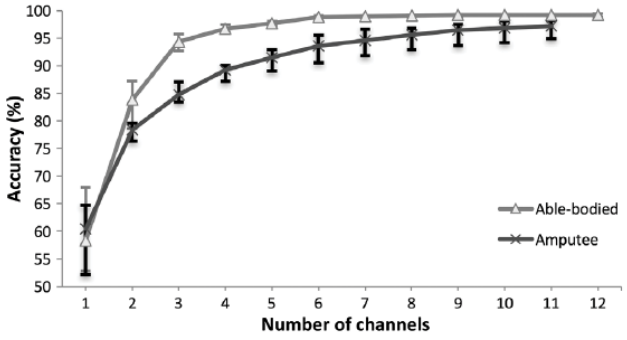


Figure 2. 38: Average accuracy versus number of channels [82]

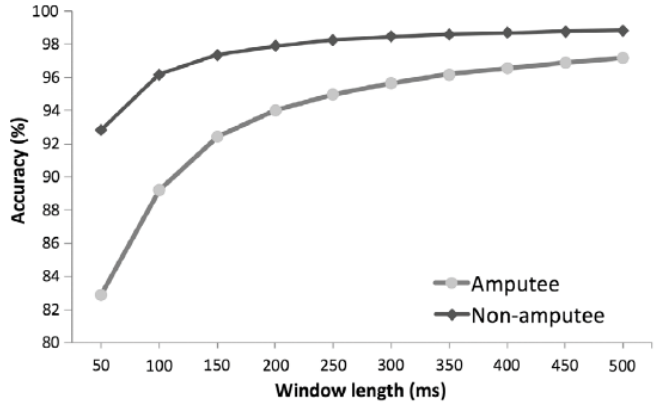


Figure 2. 39: Average accuracy against window length [82]

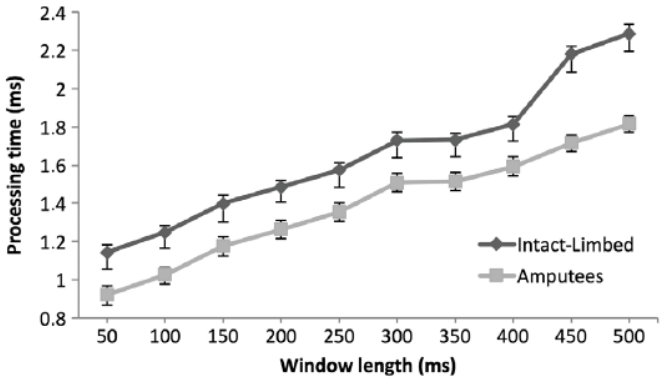
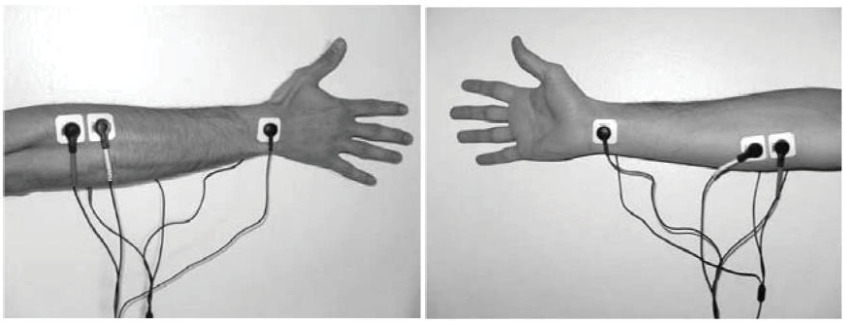


Figure 2. 40: Processing time versus window length [82]

The scholar concluded that RBF-ELM was superior to KNN and LDA while faster than both LS-SVM and SVM for the same accuracy level.

Khushaba et al. have many published contributions in the field of finger movement classification. In 2010, he was able to classify between ten finger gestures with an accuracy over 87% [23]. Here, the author used OFNDA for dimension reduction to reduce the distance between the same classes and maximise it between different classes. The researcher collected surface EMG signals from seven healthy participants by using only two electromyography channels as presented in Figure 2.41. The subjects were asked to perform ten finger gestures and each movement was iterated six times. The signal was sampled at 2000 Hz. Khushaba et al. conducted a comparative study by implementing different dimensionality reduction techniques as OFNDA, locality

preserving projection (LPP), fuzzy LDA (FLDA), orthogonal LDA (OLDA), ULDA, PCA and baseline where no dimensionality reduction applied. Figure 2.40 presents the performance associated with each dimensionality reduction algorithm. As clear from Figure 2.42 OFNDA was superior for all classes.



(a) (b)

Figure 2. 41: Electrodes allocation on right forearm in two positions (a) and (b) [23]

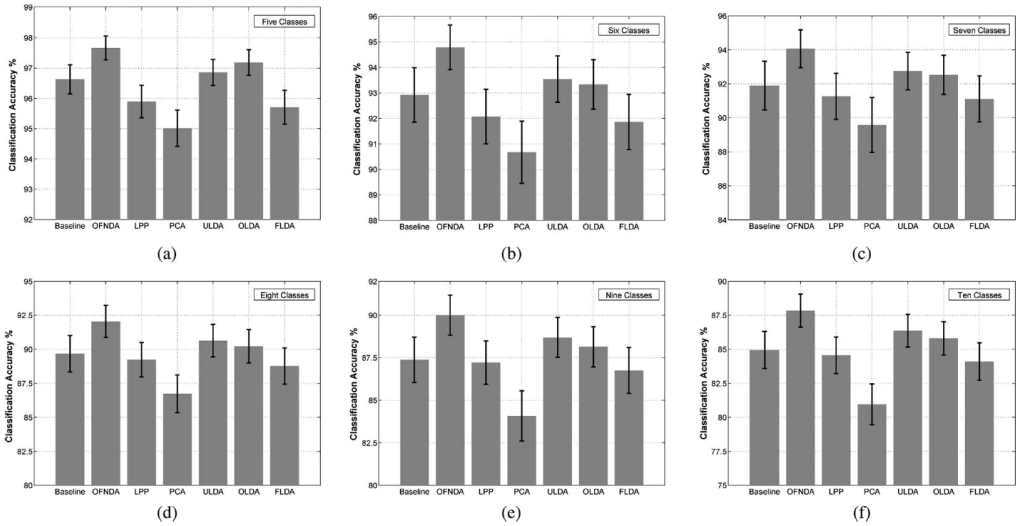
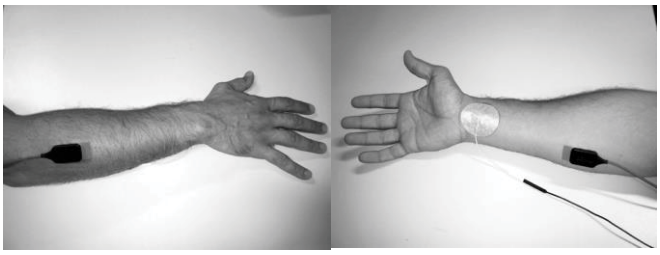


Figure 2. 42: Classification accuracy for each dimension reduction for different classes [23]

Khushaba et al. published another study, in 2011, to classify between ten finger movements by applying Fuzzy Neighbourhood Preserving Analysis (FNPA) with QR-decomposition as the feature reduction method and led to 91% accuracy with only two

channels [83]. The data were recorded from nine participants: seven males and two females. All the participating subjects were intact limbed. The recorded signal was amplified by a gain of 1000 and sampled at 4000 Hz. Moreover, the signal was filtered by a bandpass filter (20-450 Hz) and notch filter at 50 Hz. Figure 2.43 displays the placement of electrodes on the forearm in two positions whereas Figure 2.44 shows the targeted ten finger gestures.



(a) (b)

Figure 2. 43: Electrodes placement in two positions (a) and (b) [83]

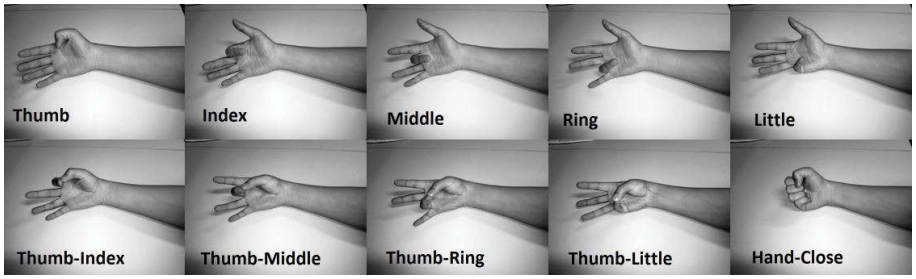
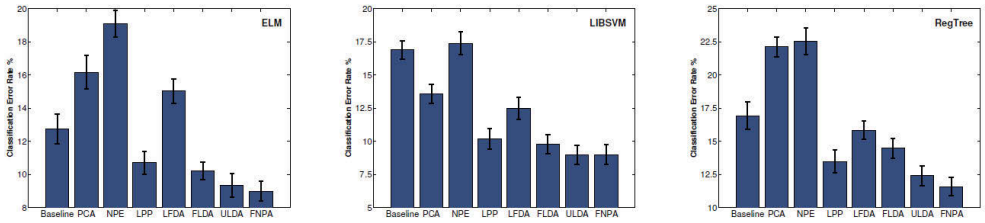


Figure 2. 44: Targeted ten finger movements [83]

Khushaba et al. utilised different classifiers such as library support vector machine (LIBSVM), Regression Tree (RegTree) and ELM. The researcher was able to achieve 91% accuracy by implementing both ULDA and FNPA as dimensionality reduction methodologies. However, ULDSA is computationally expensive while FNPA reached the same accuracy values or even better with shorter computation time. Figure 2.45 reveals the classification error achieved from implementing different dimensionality reduction methods along with three classifiers.



(a) ELM classifier (b) LIBSVM classifier (c) RegTree classifier

Figure 2. 45: Classification error attained by implementing different dimension reduction along with three classifiers [83]

Following the above-mentioned contributions in the field of finger movements classification, Khushaba et al. proposed a system that was able to classify ten finger movements from eight participants from only two EMG channels [20]. The authors recorded surface EMG signals from eight participants (six males and two females) for six trails. The first four trials were recruited for training whereas the last two were used for testing. The read signal was amplified by a gain of 1000 and sampled at 4000 Hz. Furthermore, the signal was filtered by a bandpass filter with band (20-450 Hz) and a notch filter at 50 Hz. The authors extracted many features based on previous studies such as waveform length (WL), slope sign changes (SSC) and number of zero crossings (ZC) following [84]. In addition, Hjorth time domain parameters (HTD) was extracted based on this study [85]. The dimension was reduced by applying LDA, and both SVM (LIBSVM) and KNN were applied as classifiers. Khushaba et al. applied majority vote to smooth the classifier performance. Figure 2.46 demonstrates the classification error rate for 50 ms, 100 ms and 150 ms three window length for KNN and LIBSVM classifiers.

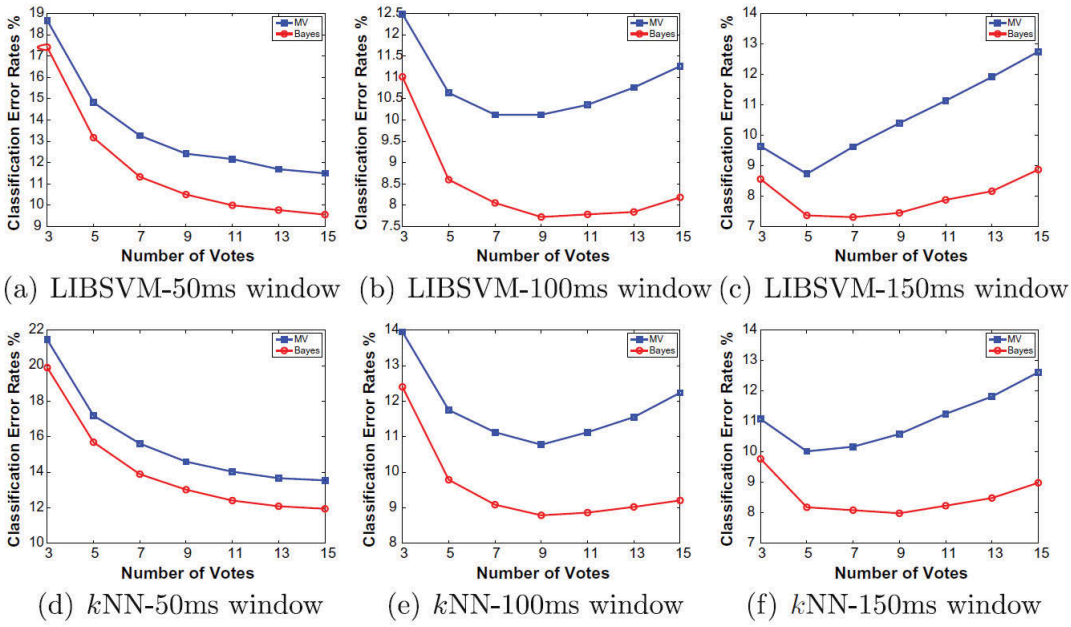
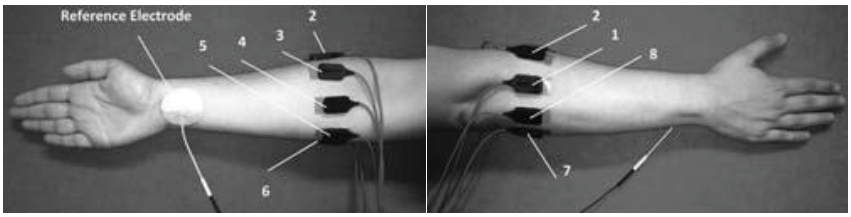
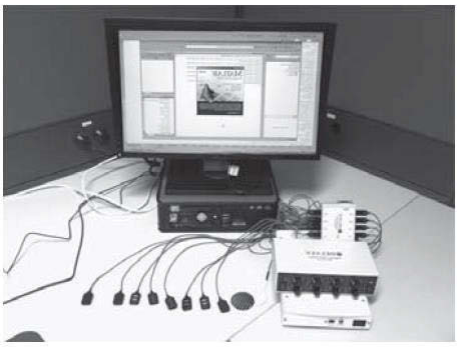


Figure 2. 46: Classification error for three different classifiers with different window sizes [20]

Other study was published by Khushaba et al. to classify fifteen finger movements [86]. The authors collected the surface EMG from eight healthy subjects by using eight channels where the signal was amplified by a gain of 1000 and sampled at frequency 4000 Hz. In addition, the collected EMG signals were filtered by bandpass filter from band (20-450 Hz) and a notch filter at frequency 50 Hz. The electrodes were allocated into interior and posterior positions as shown in Figure 2.47. Also, Figure 2.47 clarifies the hardware that was utilised to collect the signal. Figure 2.48 shows the fifteen targeted finger gestures.



(a) Electrodes allocation in interior position (b) Electrodes allocation in posterior position



(c) The used hardware in collecting the surface EMG

Figure 2. 47: Electrodes allocation at interior and posterior positions in addition to the utilised hardware in collecting the data [86]

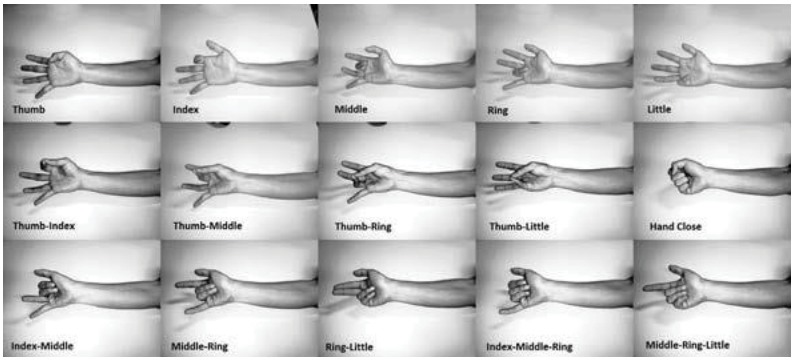
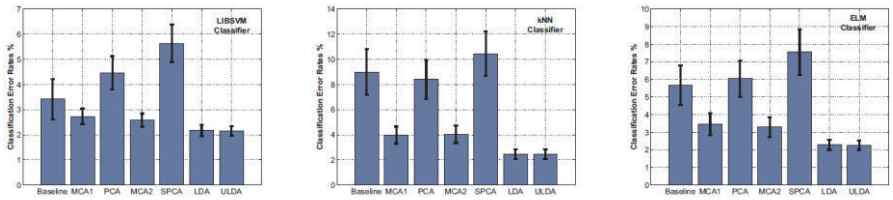


Figure 2. 48: Targeted fifteen finger gestures [86]

The time domain features were extracted and reduced using the Mutual Components Analysis (MCA) algorithm. The main difference between MCA and PCA is that MCA applies further clipping on the noise and unessential features which, in turn, led to higher accuracy values. The scholars implemented two different feature reduction techniques. The first method was based on combining PCA with mutual information (MI) and nominated MCA1 whereas the other scheme was based on Sparse PCA (SPCA) and designated MCA2. Furthermore, the authors utilised KNN, LIBSVM and ELM for classification and estimated ANOVA. Figure 2.49 presents the classification error rates for three implemented classifiers across all feature reduction implemented methodologies.



(a) Using LIBSVM classifier (b) Using *k*NN classifier (c) Using ELM classifier

Figure 2. 49: Classification error rate for three implemented classifiers with different dimension reduction techniques [86]

Lately, Khushaba et al. published a contribution in the field of finger movements where they were able to classify different finger gestures with multiple force levels in a high accuracy [87]. In this research, the authors extracted time domain features by implementing Fourier transform (FT) and recruiting norms and derivatives of the signal as mentioned in [20]. The followed feature extraction method in this study is dissimilar To either [78] or [20]. Furthermore, the researchers extracted power spectrum moments while applying normalisation to minimise the noise effect. The implemented technique was considered to be a modified version of the well-known cepstrum scheme [88]. After that, the authors estimated the cosine similarity to create a group of oriented feature sets that contained the oriented spectrum and non-linear cepstrum features. Nine amputee subjects participated in this study. Eight surface EMGs were used to acquire the signal, which was sampled at 2000 Hz. The participants were asked to perform six gestures: spherical grip, tripod grip, index flexion, thumb flexion, pine pinch and hook grip. The study was conducted at three levels of forces: low, moderate and high. It is considered a challenging task for the subjects to perform the same gesture at a different level of force. The main objective behind performing three force levels was to find a relation between the strength of the force and the amplitude of the signal. The extracted feature dimensions were reduced by employing a spectral regression (SR) scheme. Moreover, KNN, random forest (RF), naive bayes (NB) and LDA were used for classification in addition to applying majority vote. Furthermore, the ANOVA was calculated as another aspect of performance evaluation. The implementation of time

domain power spectrum achieved a 6-8% rise in the accuracy level. Figure 2.50 presents the classification error for four applied classifiers when the training and testing were performed under the same force.

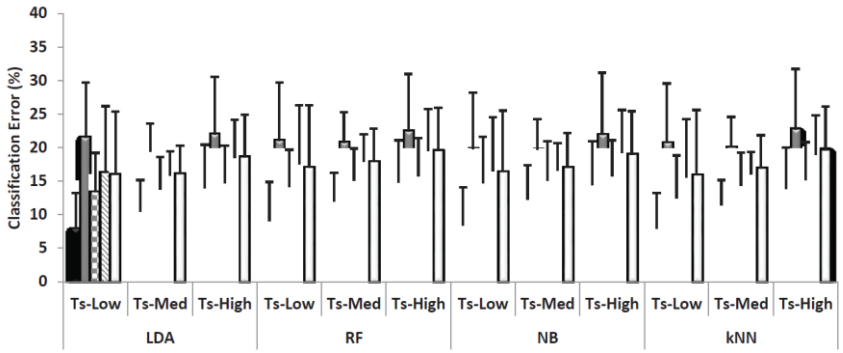


Figure 2. 50: Classification error for four classifiers under the same force [87]

Recently, the authors in [89] extracted time domain features in a different manner that caused a reduction in classification error by 8%. The researchers aimed to obtain adequate descriptors that could describe the muscle activity. Consequently, Khushaba et al. extracted power spectra from each window of the collected signal. In addition, the correlation between the extracted features for each window was estimated beside the correlation for the nonlinear mapping of the extracted features across the surface EMG channels. The same sequence was repeated for the extracted features from evert combination of the EMG channels. The authors recorded the surface EMG signal by using eleven EMG channels where nine amputee subjects participated in this study and were asked to do eleven finger movements. SR was used to apply feature dimension reduction in addition to applying both KNN and LDA for the classification task. Figure 2.51 shows the classification error rate for different feature sets by applying KNN and LDA. TSD represented Temporal-Spatial Descriptors, TDPSD denoted the previously described feature extraction algorithm that was mentioned in [87], AR-RMS signified Auto-regressive root mean square features, TD signified the mixture of mean absolute value (MAV), ZC, SSC, WL, adding Willison amplitude

(WAMP) to the same set of features led to TD1 features and finally the wavelet features. The suggested descriptors led to the lowest classification error, which was less than 5%.

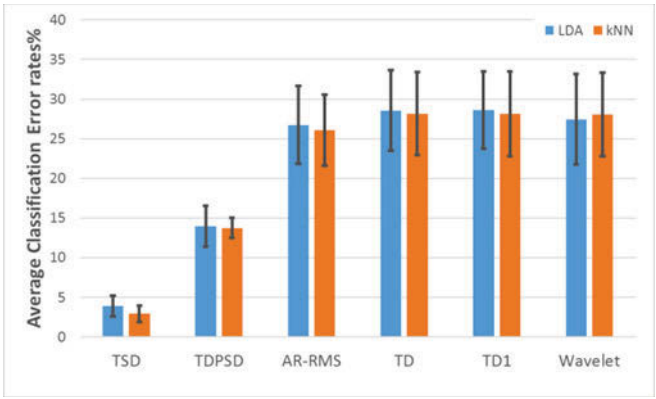


Figure 2. 51: Classification error for two classifiers with different features set [89]

We can conclude from the above presentation that the traditional architecture of any pattern recognition system that has feature extraction can be represented as shown in figure 2.52.

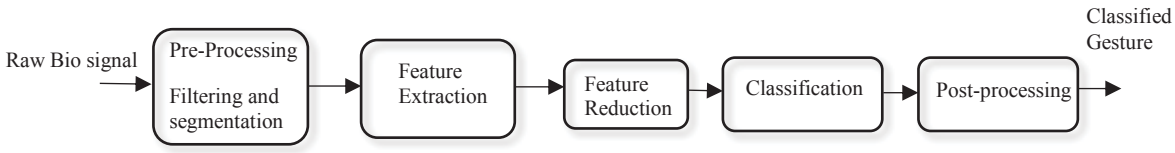


Figure 2. 52: Pattern Recognition Architecture

2.4 Biosignal Pattern Recognition

Figure 2.52 clarifies the architecture of pattern recognition as the raw biosignal moves through many phases. These stages will be simplified in brief in the following section.

2.4.1 Pre-Processing

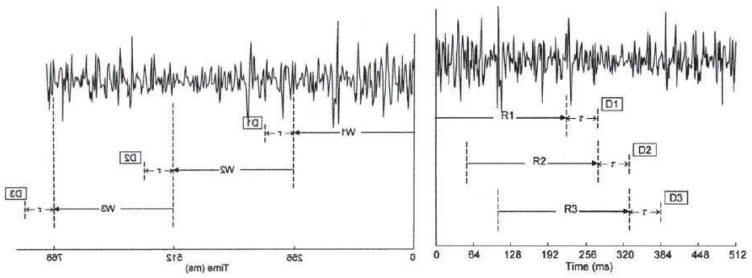
- ***Filtering***

  The main objective behind applying filters is to remove or at least minimise the existing noise. The employed filters are usually bandpass filters from 20 Hz to 450 Hz or maximum 500 Hz. The used filters can be Butterworth n^{th} order or any other type. The power produces noise at 50 Hz or 60 Hz which, requires the implementation of notch filters at 50 Hz or 60 Hz to remove the noise without eliminating the signal.

- ***Segmentation***

  The data are segmented into windows, and each window is processed individually where the valuable information (features) is extracted from each window separately. These features control the behaviour of pattern recognition model performance [52]. Basically, there are two types of windowing: either disjoint or overlapping type. Figure 2.53 clarifies the two windowing types.

  The first windowing category is the disjoint category where the window length is the only parameter as the window slides without any loss in the data. Whereas the second category is the overlapping category, which has window length and window increment where the value of increment should not exceed that of window length.



(a) (b)

Figure 2. 53: Data Segmentation via windowing (a) disjoint windowing

(b) Overlapping Windowing [90]

The electromyography signal has two states: either transient state or steady state. The transient state is the state where the generated EMG signal is produced from sudden action or movement. Kevin et al. mentioned that classification of steady-state performance was superior to that for the transient state [91]. Accordingly, most researchers consider only steady state analysis. However, analysing steady state individually makes the system not robust to any sudden changes that may exist in real applications.

2.4.2 Feature extraction

The feature extraction stage is considered to be the most vital stage in pattern recognition. The features are extracted from the processed EMG signal input. The features are characteristics that express or distinguish individual actions or classes. However, these extracted characteristics or features should have some conditions to help in achieving high accuracy rates. The first condition is to provide maximum distance between different classes which in turn increases the accuracy levels, whereas the second condition is its robustness to any noise for real-time applications. In other words, the features should not lose maximum classification ability in noisy conditions or in real time applications [21]. Phinyomark divided features types into categories [92]. The main groups are time domain features, frequency domain features and time-frequency domain features. A brief discussion of different features is presented in the following section.

- ***Time domain features***

  Time domain features are the most widely used and most accessible features [21, 22, 52]. This type of features is not computationally expensive and is easy to apply. However, they show good performance only in low noise conditions, and poor performance with transient state EMG signals [92]. Several time domain features will be explained briefly in the following section.

  1. ***Mean absolute value (MAV)***

MAV is the mean or the average for the absolute value of the collected electromyography signal during a specific segment and can be expressed as follows:

$$MAV = \frac{1}{N}\sum_{k=1}^{N} |x_k| \quad .$$

(2.16)

2. Mean absolute value slope (MAVs)

MAVs represents the difference between the MAV of the current segment and the MAV of the previous one [84]. It can be explained as follows:

$$MAVs = MAV_{K+1} - MAV_k \, .$$

(2.17)

3. Zero Crossings (ZC)

ZC counts the number of times the electromyography signal crosses the zero level [84]. A threshold is added to minimise the noise accumulation around the zero level. ZC is defined as follows:

$$ZC = \sum_{k=1}^{N} z_k, \quad z_k = \begin{Bmatrix} 1 \; x_k x_{k+1} < 0 \; and \; |x_k - x_{k+1}| \geq threshold \\ 0 \; else \end{Bmatrix}$$

(2.18

4. Slope sign changes (SSC)

SSC calculates the sum of times where the slope of the signal changes its sign [84]. It can be demonstrated as follows:

$$SSC = \sum_{k=1}^{N} z_k \, , z_k = \begin{cases} 1 \; (x_k > x_{k-1} \; and \; x_k > x_{k-1}) or \\ \quad (x_k < x_{k-1} \; and \; x_k < x_{k-1}) \; and \\ \quad |x_k - x_{k+1}| \geq threshold \;\; or \\ \quad\quad |x_k - x_{k-1}| \geq threshold \\ 0 \quad else \end{cases} .$$

(2.19)

5. ***Waveform length (WL)***

WL estimates the accumulated length of the signal during a segment. It can be represented as follows:

$$WL = \sum_{K=1}^{N} |x_k - x_{k-1}| \ .$$

(2.20)

6. ***Root mean square (RMS)***

RMS is estimated by calculating the square root of the mean of the squared segment and can be represented as follows:

$$RMS = \sqrt{\frac{\sum_{k=1}^{N} x_k^2}{N}} \ .$$

(2.21)

7. ***Autoregressive (AR)***

EMG signals are stochastic time variant or non-stationary. However, that can be considered as stationary Gaussian over a short period (segment) [93]. The AR can be estimated as follows:

$$x_k = \sum_{i=1}^{m} a_k \, x_{k-i}, \quad AR = a_k \quad m^{th} \, order \, AR \, model \ .$$

(2.22)

8. ***Hjorth parameters***

The Hjorth features extracts three parameters from either EEG or EMG signals: activity, mobility and complexity [20, 94]. The three parameters can be defined as follows:

$$Activity: \ m_0 = var(x(t))$$

$$Mobility: \ m_1 = \sqrt{\frac{m_0(\frac{dx(t)}{dt})}{m_0(x(t))}}$$

$$Complexity: \ m_2 = \frac{m_1(\frac{dx(t)}{dt})}{m_1(x(t))},$$

(2.23)

where $x(t)$ is the EMG signal or any other signal.

The above-mentioned eight time domain features are the most popular and widely used. In the following section, frequency domain features will be demonstrated briefly.

- ***Frequency domain features***

  Frequency domain features are mostly used in analysing fatigue, estimated from the power spectral density (PSD). Two favorite frequency domain features are mean frequency and median frequency. These two features will be demonstrated briefly in the following section.

  1. ***Mean frequency (MNF)***

     MNF is estimated by calculating the summation of the multiplication of the current frequency multiplied by its power spectral density over whole segments and divided by the summation of power spectral density across entire segments. The mean frequency or average frequency can be explained as follows:

$$MNF = \sum_{k=1}^{N} f_k P_k \Big/ \sum_{k=1}^{N} P_k \,,$$

(2.24)

     where f_k represents the frequency spectrum while P_k is the power spectrum at the frequencyf_k.

  2. ***Median frequency (MDF)***

     MDF calculates the frequency at which the power spectrum of the signal is separated into two areas, each of the same amplitude. MDF can be estimated by the following formula:

$$\sum_{k=1}^{MDF} P_k = \sum_{k=MDF}^{N} P_k = \frac{1}{2}\sum_{k=1}^{N} P_k \,.$$

(2.25)

- ***Time-Frequency domain features***

  Time-Frequency domain (TFD) features lead to better results and higher accuracy. The main reason behind TFD leading to better performance is its ability to confine the signal energy in both domains, either time domain or frequency domain [95]. However, the only drawback is its complexity and simulation time as it is considered computationally expensive, which makes it not suitable for real-time applications. Wavelet packet transform (WPT), wavelet transform (WT) and short-time Fourier transform (STFT) are considered the most widely used TFD features [91]. Figure 2.52 shows the three leading TFD features where the x-axis represents time, and the y-axis signifies the frequency. As is clear in Figure 2.54, SFTT has a fixed tiling whereas WT has a tiling aspect ratio that is related to the centre frequency. WPT has an adaptable tiling, which makes it suitable for most of the applications, yet it consumes longer time and is computationally expensive, which in turn has a bad impact on real-time applications. More detail will be presented on TFD features in chapter 3 in this thesis.

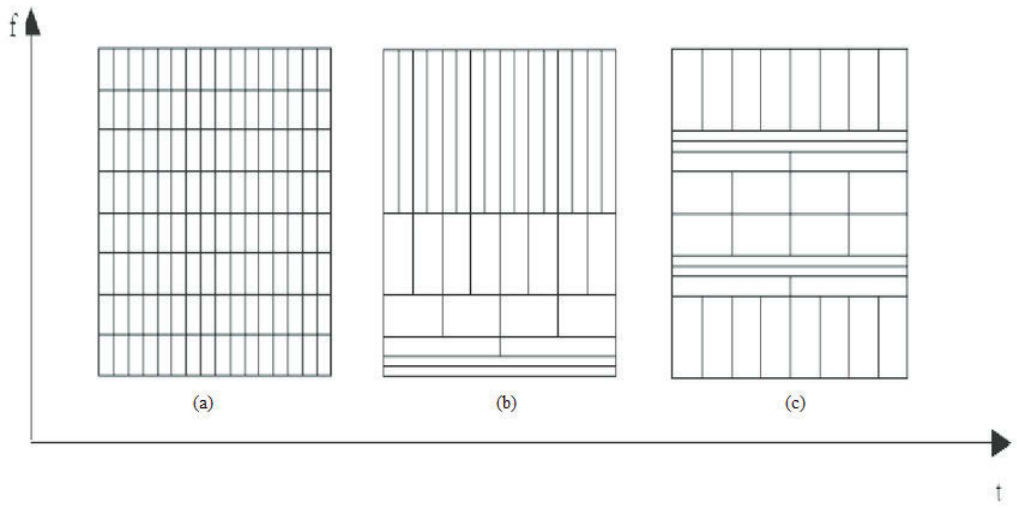


Figure 2. 54: Three leading TFD features (a) STFT (b) WT (c) WPT [95]

2.4.3 Feature reduction

Feature reduction is an essential stage in the pattern recognition block diagram shown in Figure 2.52. When features are extracted, this results in a large number of features that may be redundant. Therefore a technique may be utilised to reduce the number of features without losing any information that may enhance the performance of the system. In general, feature reduction can be accomplished by either feature selection (FS) or feature projection (FP) [95]. Feature selection algorithms select a subset of features from the extracted features whereas feature projection transforms or maps the extracted features from its original space to a new space of lower dimension. Based on the studies conducted in [23, 95], the authors demonstrated that feature projection achieved better performance than implemented feature selection techniques. More analysis will be demonstrated on feature selection later in this chapter. This feature projection methodology has two schemes either unsupervised or supervised. The unsupervised scheme maps the features from a higher dimension to the new space of lower dimension without containing any information about the classes. On the other hand, the supervised scheme also transforms the extracted features from a higher dimension to a lower dimension whereas including information on the classes. Principle component analysis (PCA) is an illustration of unsupervised feature projection [96], and LDA is an example of supervised projection [97]. Supervised feature projection methods showed superior performance better than include information on the class during transformation [22]. The significant disadvantage of LDA in specific is the singularity problem, which arises when the number of samples is lower the number of classes. Researchers suggested various supervised feature projection methods to beat the singularity problem, such as ULDA [98], SRDA [97], and OFNDA [23]. Moreover, feature projection can be either linear or nonlinear. PCA and LDA are examples of linear feature projection with replacing the linear function in PCA by the kernel function lead to a nonlinear version of PCA. Additionally, neural

networks such as autoencoders and unsupervised extreme learning machines (USELM) are good examples of nonlinear feature projection [99].

PCA, LDA and SRDA is presented briefly in the following sections

- ***Principle component analysis (PCA)***

  PCA is an algorithm to reduce the dimension of the data by mapping the original data set into a lower dimensional space while preserving vital information [100]. The PCA technique is based mainly on estimating the principal components from the original signal by applying linear combination between them.

  Assume that we have a dataset $X = \{x_1, x_2, \dots\dots, x_N\}$ where X is the original dataset which exists in a t-dimensional space and N represents the number of samples. PCA applies linear combination between lower dimension data and keeps the important information along with mapping into a lower m-dimensional space. Assume that P describes the transformation from higher dimension space t to lower dimension m, where $m < t$. Therefore, the new dataset of lower dimension will be represented by $y_i \in R^m$. The y_i value can be estimated via the following formula.

  $$y_i = P^T x_i \quad i = 1, \dots, N \,, \tag{2.26}$$

  where $x_i \in R^t$ and P contains eigenvectors e_i which are represented in columns. Singular value decomposition (SVD) is the guide to estimate the eigenvalues as follows:

  $$\lambda_i e_i = Q e_i \tag{2.27}$$

  $$Q = XX^T \,, \tag{2.28}$$

  where λ_i represents the eigenvalues related to the eigenvector e_i, while Q signifies the covariance matrix.

- ***Linear discriminant analysis (LDA)***

LDA does not target the dataset or features. However, it targets the classes as it looks for the best discriminant for classes and not the best features, as in PCA [96]. The best discriminant is achieved through minimising the scattering matrix within the same class while maximising it between different classes. The scatter matrix for the same class can be described via the following equation

$$S_w = \sum_{j=1}^{c} \sum_{i=1}^{N_j} (x_i^j - \mu_j)(x_i^j - \mu_j)^T \,, \tag{2.29}$$

where x_i^j represents the i^{th} sample in the j^{th} class, μ_j is the mean of the j^{th} class, c describes the total number of classes and N_j defines the total number of samples in the j^{th} class.

On the other hand, the scattering matrix that differentiate between different classes can be described as follows:

$$S_b = \sum_{j=1}^{c} (\mu_j - \mu)(\mu_j - \mu)^T \,. \tag{2.30}$$

In order to achieve LDA objective, minimise the scatter matrix within the same class and maximise it for different classes, this can be achieved by maximising the LDA objective function as follows:

$$a^* = arg_a max \frac{a^T S_b a}{a^T S_w a} \,. \tag{2.31}$$

In case of a non-singular matrix S_w, the objective function will be maximum when the projection matrix P column vectors are identical to the eigenvectors $S_w^{-1} S_b$.

Otherwise, many researchers proposed different techniques to overcome the singularity problem in the case of the singularity problem in LDA as mentioned above.

- ***Spectral regression discriminant analysis (SRDA)***

SRDA is more superior to LDA, especially with a more extensive dataset [97]. Moreover, it was implemented as a solution to the singularity problem that may arise in LDA. The LDA problem can be overcome by the following analysis.

$$\overline{X}W\overline{X}^{T}a = \lambda\overline{X}\overline{X}^{T}a \ ,$$

(2.32)

where $\overline{X}$ represents the data matrix of dimension $(1xc)$, where c is the total number of classes, W describes the eigenvector matrix of dimension (mxm), m is the total number of data, λ define the eigenvalue, and a designates the transformation vector. Equation (2.32) can be simplified as follows:

$$W\overline{y} = \lambda\overline{y} \ ,$$

(2.32)

where $$\overline{y} = \overline{X}\overline{X}^{T}a \ .$$

(2.33)

To solve these equations, the values of $\overline{y}$ need to be found in equation (2.32). Also, the value of a that verifies equation (2.33) should be estimated via least square method. This led to the following formula

$$a = arg_a min \sum_{i=1}^{m} \left(a^T\overline{x}_i - \overline{y}_i\right)^2 \ .$$

(2.34)

By rearranging the SRDA least square problem, the following equation can be estimated:

$$= arg_a min \sum_{i=1}^{m} (\left(\overline{X}^T a - \overline{y}\right)^T \left(\overline{X}^T a - \overline{y}\right) + \alpha\, a^T a) \ .$$

(2.35)

The derivative of equation (2.35) led to the following formula

$$\left(\overline{XX}^T + \alpha I\right) = \overline{X}\,\overline{y}\,.$$

(2.36)

This led to the following value for a:

$$a = \left(\overline{XX}^T + \alpha I\right)^{-1} \overline{X}\,\overline{y}\,.$$

(2.37)

2.4.4 Classification

Classification is the final stage in pattern recognition in case of omitting the post-processing stage. The classifier maps the reduced features into classes where each feature is translated to the class to which the feature belongs. Over the past years, many researchers implemented different classification techniques such as in [101] where a multilayer perceptron (MLP) was utilised for classification. In an earlier research, the authors of [102] applied feed-forward neural networks (FNN) for classification. FNN showed superior performance but since it is still considered to be a neural network which needs training time. So, it is computationally expensive and will not be suitable for the real-time applications. Later, the researchers adopted the idea of implementing LDA for classification [52]. The results were impressive as the accuracy was higher than MLP and as precise as FNN but with shorter computation time. Moreover, KNN and hidden Markov model (HMM) was suggested for pattern recognition [103] and [104] respectively. Lately, SVM was used in many applications, which led to promising results as it showed superior performance than that for either KNN, FNN or LDA and consumed relatively little computational time. However, the only drawback for SVM was that it was initially used for classifying two classes only. Modifications were introduced to SVM to make it suitable for multi-class applications, which in return increased the computational time. Recently, ELM was presented where the idea came

from artificial neural networks [105]. However, ELM does not follow the traditional iterative learning to tune the node parameters, which in turn makes it faster in implementation than the conventional networks. Two types of ELM were suggested [105]. The first kind was kernel-based ELM, where its performance mainly depended on the type of the implemented kernel function like SVM, whereas the other kind was node-based ELM. ELM, SVM, DA and softmax layer classifiers will be demonstrated in the following section.

- ***Extreme learning machine (ELM)***

  Extreme learning machines were first introduced in 2012 by Huang [105]. The first issuance for ELM was not as a classifier. However, it was just a modification on single-hidden-layer feed-forward networks (SLFNs). The authors suggested omission of the iterative methodology in estimating the weights in either the output or in the hidden layers. This technique made the weights in the hidden layer entirely independent on the input data which, in turn, made ELM adaptable to any data type. The weights for the output layer were computed by applying Moore-Penrose inverse to the hidden layer matrix. This technique made ELM much faster than conventional SLFNs. Furthermore, ELM resulted in high accuracy values compared to other classification algorithms, which were considered to be computationally expensive. However, Hunag et al. were not the first researchers to suggest avoiding iterative estimation of the weights. Randomisation was proposed before by Pao and Takefji in [106]. The authors suggested a single-hidden-layer feed-forward network, where the hidden layer was added to the input layer, and called it functional-link net (FLN). Later, in 1994 Pao et al. applied an adjustment to FLN where the weights of the added hidden layer are estimated randomly and called it the random vector functional-link net (RVFLN) [107]. ELM and RVFLN are similar as both follow the random manner for estimating weights. However, RVFLN merged the hidden layer with the input layer whereas ELM kept the conventional structure of SLFNs.

Assume that we have a fundamental single layer neural network as shown in Figure 2.55. Input data are represented by $x_j = [x_{j1}, x_{j2,\ldots\ldots,} x_{jm}]^T \in R^m$. Equivalent output classes or targets are signified $y_j = [y_{j1}, y_{j2,\ldots\ldots,} y_{jm}]^T \in R^n$.

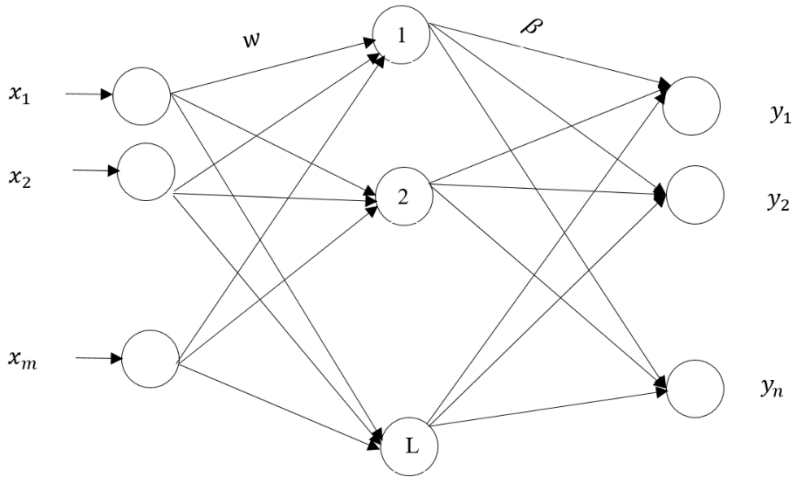


Figure 2. 55: The architecture of primary extreme learning machine network

The output from a standard single-hidden-layer feed-forward network, where L represents the number of neurons in the hidden layer, can be defined as follows:

$$f_j(x) = \sum_{i=1}^{L} \beta_j \, g\left(w_i . x_j + b_j\right) = G\beta \qquad J = 1, \ldots., N \,,$$

(2.38)

where w_i defines the weight that connects the i^{th} neuron and input, β_i describes the weight that connects the i^{th} neuron and output, b_i expresses the bias and $g(x)$ signifies the hidden node activation function. G and β can be further expressed in detail in the as following:

$$G = \begin{bmatrix} g(w_1 . x_1 + b_1) & \cdots & g(w_L . x_1 + b_L) \\ \vdots & \ddots & \vdots \\ g(w_1 . x_N + b_1) & \cdots & g(w_L . x_N + b_L) \end{bmatrix}.$$

(2.39)

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_1^T \end{bmatrix}.$$

(2.40)

Besides minimising the error, ELM aims to minimise the norm that estimates the output weights as well. This can be expressed via the following formula

$$min_{w_i b_i \beta}: ||G(w_1, \dots, w_L, b_1, \dots, b_L)\beta - T||^2 ,$$

(2.41)

where T represents the output classes or the targets.

$$T = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix}.$$

(2.42)

As in ELM, both the input weights w_i and input bias b_i are estimated randomly so that equation (2.41) can be reduced to

$$min_{\beta}: ||G\beta - T||^2 .$$

(2.43)

The minimum norm for equation (2.43) can be computed by implementing the Moore Penrose generalised inverse for the matrix G, which is clarified via the following formula:

$$\beta = G^{\dagger}T .$$

(2.44)

The optimisation for ELM is done as follows

$$min: L_{PELM} = \frac{1}{2}||\beta||^2 + C\frac{1}{2}\sum_{i=1}^{N}||\xi_i||^2 .$$

(2.45)

$$g(x_i)\beta = t_i^T - \xi_i^T \quad i = 1, \dots . N ,$$

(2.46)

where $\xi_i^T = [\xi_{i,1}, \dots \dots, \xi_{i,n}]$ represents the output error vector and $g(x)$ is feature mapping from the hidden layer to the output. Following the theorem mentioned in

[108] it was noted that ELM training was considered as the solution for dual optimisation problem for equation (2.45). Therefore

$$min: L_{DELM} = \frac{1}{2}||\beta||^2 + C\frac{1}{2}\sum_{i=1}^{N}||\xi_i||^2 - \sum_{i=1}^{N}\sum_{j=1}^{n}\alpha_{i,j}\ (g(x_i)\beta - t_i^T + \xi_i^T)\ ,$$

(2.47)

where β_j denotes the weights linking the hidden layer with the output layer and C is a parameter to be defined by the user. The differentiation for equation (2.47) will be estimated as follows:

$$\frac{\partial L_{DELM}}{\partial \beta_j} = 0\ \rightarrow \beta_j = \sum_{j=1}^{N}\alpha_{i,j}g(x_i)^T \rightarrow \beta = G^T\alpha\ .$$

(2.48)

$$\frac{\partial L_{DELM}}{\partial \xi_j} = 0\ \rightarrow \alpha_i = C\xi_i\ .$$

(2.49)

$$\frac{\partial L_{DELM}}{\partial \alpha_i} = 0\ \rightarrow g(x_i)\beta - t_i^T + \xi_i^T = 0\ .$$

(2.50)

By replacing equation (2.48) by both (2.49) and (2.50), we get

$$\left(\frac{1}{C} + GG^T\right)\alpha = T\ ,$$

(2.51)

and

$$T = \begin{bmatrix} T_1^T \\ \vdots \\ T_N^T \end{bmatrix} = \begin{bmatrix} t_{11} & \cdots & t_{1n} \\ \vdots & \ddots & \vdots \\ t_{N1} & \cdots & t_{Nn} \end{bmatrix}.$$

(2.52)

Replacing the value of T from equation (2.44) by that in equation (2.51), we obtain

$$\beta = G^T(\frac{1}{C} + GG^T)^T\ .$$

(2.53)

Finally, the output of ELM can be defined via the following equation:

$$f(x) = g(x)\beta = g(x)\, G^T(\frac{1}{C} + GG^T)^{-1}T \ .$$

(2.54)

Equation (2.54) can be rewritten as

$$f(x) = g(x)\beta = g(x)\, G^T(\frac{1}{C} + GG^T)^{-1}G^TT \ .$$

(2.55)

The feature mapping function $g(x)$ can be expressed in terms of $Q(w, b, x)$. The feature mapping function can be any nonlinear continuous function such as a multi-quadratic function, hard limit, sigmoid, and Gaussian. The formula for each function will be presented in the following equations:

Multi quadratic function

$$Q(w, b, x) = \ (\left|\left|x - w\right|\right|^2 + b^2)^{1/2} \ .$$

(2.56)

Hard limit function

$$Q(w, b, x) = \begin{cases} 1, if\ (w.x - b) \geq 0 \\ 0,\ otherwise \end{cases} .$$

(2.57)

Sigmoid function

$$Q(w, b, x) = \frac{1}{1+\exp(-(w.x-b))} .$$

(2.58)

Gaussian function

$$Q(w, b, x) = \exp(-b||x - w||^2) \ .$$

(2.59)

The mapping function, which is enabled in the hidden layer, can be replaced by a kernel function and in this case, the ELM will be a kernel based ELM. The kernel matrix can be denoted as follows:

$$\Omega_{ELM} = GG^T\Omega_{ELMi,j} = g(x_i).g\left(x_j\right) = k(x_i, x_j) \ ,$$

(2.60)

where $k(x_i, x_j)$ is the kernel function. Equation (2.54) can be rewritten as

$$f(x) = g(x)\, G^T \left(\frac{I}{C} + GG^T\right)^{-1} T \ .$$

(2.61)

$$f(x) = \begin{bmatrix} k(x, x_1) \\ \vdots \\ k(x, x_N) \end{bmatrix}^T \left(\frac{I}{C} + \Omega_{ELM}\right)^{-1} T \ .$$

(2.62)

The kernel function can be a linear, polynomial or radial basis function. The three types will be illustrated as follows

Linear

$$k(x_i, x_j) = x_i . x_j \ .$$

(2.63)

Polynomial

$$k(x_i, x_j) = (x_i . x_j + 1)^d \ .$$

(2.64)

Radial basis function

$$k(x_i, x_j) = exp(-\gamma ||x_i - x_j||) \ .$$

(2.65)

- ***Support vector machine (SVM)***

The support vector machine (SVM) is a widely used classification algorithm, first developed in 1995 by Cortes and Vanpik [109]. Support vector machines initially considered classifying two classes only. Therefore, SVM aimed to search for the best hyperplane that actively separated two different sets of data points. Accordingly, SVM

created a width, which was called a margin to separate the data. The wider the margin width, the greater the capability of the classifier. The data points that are located on the borders of the hyperplane are called the support vectors. Figure 2.56 shows the support vector machine classifier including hyperplane and support vectors.

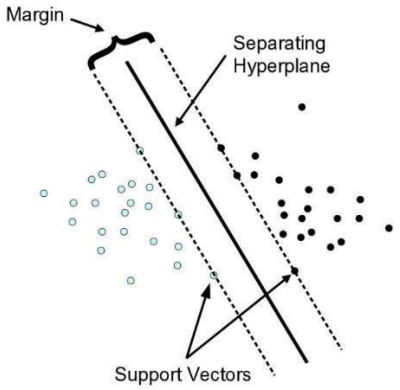


Figure 2. 56: Support vector machine classifier [110]

Many data sets cannot be separated entirely, which results in a misclassification. In other words, the data points will be classified on the wrong side of the hyperplane. The influence of these data points will be reduced by weighting them down via a soft margin technique [111]. In some cases, there may be no suitable linear separator that can separate between data sets which lead to projecting the current datasets into a new higher dimensional space by employing kernel method [112]. The mathematical analysis for support vector machines can be found in detail in [113, 114].

Assume that we have a training set in the form of vectors called x_j with equivalent categories y_j. In the dimension, $x \in R^d$ and $y_j = \pm 1$. The hyperplane achieves $f(x) = x'.\beta + b$ =0 where $\beta \in R^d$ and b is a real number. SVM aims to maximize the margin which happens when $||\beta||$ is reduced to its minimum extent for all values $(x_j, y_j), y_j f(x_j) \geq 1$. The support vectors are the values of x_j that are positioned on the boundary of the hyperplane and achieve $y_j f(x_j) = 1$. $||\beta||$ is reduced to its maximum extent in case of $class(z) = sign(z'\hat{\beta} + \hat{b})$= $sign(\hat{f}(z))$ where z is the

distance to the hyperplane and $\hat{f}(z)$ is the classification value. By using a LaGrange multiplier, we get the following formula:

$$L_P = \frac{1}{2}\beta'\beta - \sum_j \alpha_j \left(y_j \left(x_j' \beta + b\right) - 1\right) .$$

(2.66)

For stable values of both β and b, estimating the derivative of L_P and equating to zero we get:

$$L_D = \sum_j \alpha_j - \frac{1}{2} \sum_j \sum_k \alpha_j \alpha_k y_j y_k x_j' x_{k'} .$$

(2.67)

Some data sets cannot be separated. A variable ξ_j is added and cost parameter c is included.

$$L_P = \frac{1}{2}\beta'\beta + c\sum_j \xi_j - \sum_j \alpha_j \left(y_j f\left(x_j\right) - (1 - \xi_j) - \sum_j \xi_j \mu_j\right) ,$$

(2.68)

Where μ_j is the Lagrange Multiplier

In the case of not being able to find a linear hyperplane, a kernel function is employed to map the current datasets into a new higher dimensional space. The new space is s where the classifier aims to map the data set x into it. The class function is expressed as $G(x_1, x_2)$. The class function can be

Polynomial $G(x_1, x_2) = (1 + x_1' x_2)^p$, where p is a positive integer

(2.69)

Gaussian $G(x_1, x_2) = \exp(-||x_1 - x_2||^2)$.

(2.70)

Sigmoid Neural Networks $G(x_1, x_2) = \tanh(p_1 x_1' x_2 + p_2)$, where p_1 is for positive integers and p_2 is for negative integers.

(2.71)

SVM was implemented as a classification algorithm in different fields. For example, SVM was employed in the civil engineering field to classify features in model induction where the authors compared the results from SVM by those resulting from artificial neural networks [115]. SVM was used in image classification field for both labelled and unlabeled datasets. The authors in [116] suggested using SVM to classify unlabeled data sets to decide which text should be selected next. Other researchers implemented SVM to classify between bullet's image [117]. SVM was able to classify between images of defects bullets and images of non-defects bullets. The proposed algorithm reached 90% as an accuracy level for 80 images where 40 of them were with defects and other 40 were without defects.

- ***Softmax layer***

Softmax classification is an adapted version of logistic functions or binary regression classifiers to fit classifying multiclass applications [118]. The loss function for a softmax classifier is different from that of SVM. Softmax classifiers generate the normalised probability for each class. The loss function for softmax classifier can be expressed as a mean square error, or cross entropy. Mean square error can be expressed as:

$$E = \frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{k}(t_{ij} - y_{ij})^2 ,$$

(2.72)

where n is the number of training samples, k is the number of training classes, t_{ij} is the ij^{th} entry of target matrix, target matrix is k by n and y_{ij} is the i^{th} output from autoencoder with input vector x_j.

A cross entropy loss function is defined as:

$$E = \frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{k} t_{ij} ln y_{ij} + \big(1 - t_{ij}\big)\ln(1 - y_{ij}) .$$

(2.73)

The softmax classifier has been employed in many applications, especially those associated with biosignals. Besides, neural networks either for deep learning or traditional feature extraction and reduction algorithms. The authors in [119] proposed implementing a softmax classifier to classify Red Green Blue (RGB) colour images, and proved a higher classification ability than that of neural networks. Other researchers used softmax classifiers to detect drugs where the classification result was higher than the well-known state of the art techniques [120].

- ***Discriminant analysis***

Discriminant Analysis classification is a widely used classifier technique which depends on Fisher's discriminant. It looks for a mix of features that differentiate between different classes, either two or more [121]. Discriminant analysis can be used in dimension reduction as well. The commonality between discriminant analysis and Analysis of Variance (ANOVA) is that both of them look for a mix of features [122]. The only difference between ANOVA and discriminant analysis is in the type of variable, wherein ANOVA the variable is of categorical type while in the discriminant analysis the variable is of continuous-type [123]. In addition, Principal Component Analysis is similar to Linear Discriminant Analysis as both are looking for a linear combination of features which best express the data [96]. The mean of the classes for unweighted data is expressed as follows

$$\hat{\mu}_k = \frac{\sum_{n=1}^{N} M_{nk}\, X_n}{\sum_{n=1}^{N} M_{nk}}$$

(2.74)

Assume that we have matrix M with dimensions N by K where N represents data and K represents classes.

$M_{nk} = 1$ if data n is from class K

$M_{nk} = 0$ Otherwise

The mean of the class for positively weighted data w_n is:

$$\hat{\mu}_k = \frac{\sum_{n=1}^{N} M_{nk}\, X_n\, w_n}{\sum_{n=1}^{N} M_{nk}\;\; w_n}$$

(2.75)

The value of the covariance matrix for unweighted data can be estimated as follows:

$$\hat{\Sigma} = \frac{\sum_{n=1}^{N}\sum_{k=1}^{K} M_{nk}\,(X_n - \hat{\mu}_k)(X_n - \hat{\mu}_k)^T}{N-K}.$$

(2.76)

The covariance matrix for weighted data w_n can be expressed as follows

$$\hat{\Sigma} = \frac{\sum_{n=1}^{N}\sum_{k=1}^{K} M_{nk}\, W_n (X_n - \hat{\mu}_k)(X_n - \hat{\mu}_k)^T}{1-\sum_{k=1}^{K}\frac{W_k^{(2)}}{W_k}},$$

(2.77)

where $W_k = \sum_{n=1}^{N} M_{nk}\, W_n$ represents the sum of weights per class,

and $W_k^{(2)} = \sum_{n=1}^{N} M_{nk}\, W_n^2$ is the sum of square of weights for each class.

The discriminant analysis can be estimated by applying cost, posterior and prior probabilities:

$$\hat{y} = \arg min_{y=1,\ldots\ldots K} \sum_{k=1}^{K} \hat{P}(k|x) C(y|k)\ ,$$

(2.78)

where $\hat{y}$ represents the final predicted class, K represents the number of classes, $\hat{P}(k|x)$ represents the posterior probability for data x to be from class k and $C(y|k)$ represents the cost of classifying y as an observation when its right class is k.

The posterior probability of an observation x that belongs to a class k is the multiplication of normal density and prior probability which is represented as follows

$$P(x|k) = \frac{1}{(2\pi|\Sigma_k)^{1/2}} \exp(\frac{-1}{2} (X_n - \mu_k)^T \sum_k^{-1} (X_n - \mu_k)),$$

(2.79)

where μ_k is the mean of the normal density of observation x, Σ_k is the covariance of normal density of observation x and Σ_k^{-1} is the inverse matrix.

The posterior probability of an observation x is of class k and $P(k)$ represents the prior probility of class k:

$$\hat{P}(K|x) = \frac{P(x|k)P(k)}{P(x)}.$$

(2.80)

Discriminant analysis is a very popular classifier algorithm and many researchers published its application as a classifier. The authors of [124] suggested many advanced schemes in discriminant analysis like linear, kernel and Monte Carlo applications. Other investigators developed a study that compared between different discriminant analysis based on collected gene data to classify the patients with cancer [125]. Other researchers introduced sparsity to discriminant analysis by applying rotation techniques on data which achieved enhanced results [126].

2.4.3 Post-processing

The post-processing stage is the last stage in pattern recognition, which aims to smooth and improve the classifier results. The majority vote is one of the most widely used post-processing techniques [90]. The majority vote estimates the classification by considering both current state and n previous states, which in turn increases accuracy from one side as it removes the misclassified results [127]. However, from the other side majority vote increases the computation time, which should bring attention to compromise between rising accuracy levels and limiting delay. The delay should be in

line with the endorsement [128]. whereas the expected delay in real time applications should range between (100-150 ms) or at least does not exceed 300 ms [91].

2.5 Summary

A very fundamental study was presented on hand anatomy including bones, skin, nerves, muscles, tendons and joints. Then a brief theory was presented on biosignal in general and EMG in particular. This theory included skin preparation for EMG collection, factors affecting EMG signal and EMG applications.

Then a discussion was conducted including the researchers' efforts in the field of finger movements classifications by employing different suggested pattern recognition models. Some of these discussed models achieved superior results, however they did not classify many finger gestures as the maximum number of classified finger movements was around 15 classes only. In the laboratory the classified finger gestures were much more than 15 gestures, which in turn created a gap between the simulation results and real-life applications results. Therefore, the simulation results were superior when it came to the laboratory the results decreased dramatically. This decay due to many results included but not limited to the robustness of EMG signal and in clinical applications, the finger gestures are unlimited while in simulation the tested movements were limited. Later, an elementary study was shown of the stages of a pattern recognition system with an emphasis on features, either time or frequency domain. Finally, different feature selection techniques were discussed and applied our dataset.

This thesis will give attention to the feasibility of learning features for electromyography signals. The thesis will present different techniques to implement deep learning along with signal representation to achieve better results with less human intervention. Chapter 3 will discuss the concept of feature selection, and deep learning. Neural networks and different signal representations algorithms will be shown. Chapter 4, Chapter 5, chapter 6 and chapter 7 will display and discuss the proposed and

published feature and deep learning models that were accomplished during this study. Eventually, Chapter 8 will have the thesis summary, recommendations for future work and conclusion.

CHAPTER 3

Review on feature learning, deep learning and data representation

The principal objective of this research is to develop a deep learning model that learns features by itself without employing any feature extraction or reduction techniques or even hiring handcrafted engineering methods. Self-learning models will not depend on user's experience in extracting most suitable features to the application. Hence, it will save time and effort wasted in iteration trials to pick up the best features. This research investigates different feature learning and deep learning algorithms that lead to the ultimate benefits of the learnt features.

In this chapter, an in-depth study will be presented on feature learning, deep learning as well as the neural network as the neural network is one of the commonly implemented methods to achieve deep learning concept. Various signal representation algorithms that were recalled in our proposed models, will be illustrated in this chapter. The signal representation approaches will be spectrogram, wavelet, wavelet packet, data augmentation, and tensor. Furthermore, classifier fusion will be demonstrated as classifier optimisation technique to select best local classifier.

3.1 Feature selection

This thesis is more concerned in learning the best features that match the application, which will be presented in the coming chapters as feature learning, and deep learning methodologies. Nevertheless, before proceeding to the deep learning area some brief introduction to traditional feature selection algorithms will be given.

Feature selection is the process of selecting a subset of features from a pool of features. Feature selection is sometimes called variable selection, attribute selection or variable subset selection. The technique for feature selection is useful as it reduces training time, simplifies the model to be more comfortable for researchers to understand [129] and reduces variance by reducing overfitting [130] as it lowers complexity by decreasing

parameters relative to the number of observations. A typical feature selection process consists of four necessary steps as shown in Figure 3.1. The four necessary steps are subset generation, subset evaluation, stopping criterion, and result validation [131]. Figure 3.1 clarifies the very general and necessary steps for any feature selection model. These steps will be briefly demonstrated in the following section.

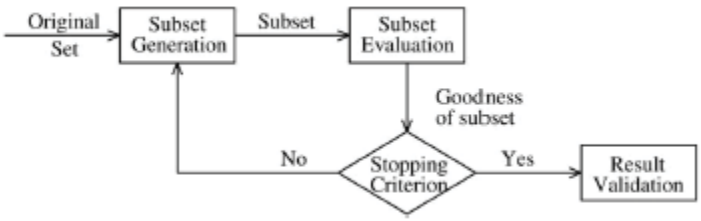


Figure 3. 1: Four key steps of feature selection [131]

3.1.1 Subset Generation

Subset generation is the first process in feature selection procedures. Many search strategies can be used to pick a subset of features then, the selected subset is evaluated using a satisfied evaluation criterion to help in taking the decision. The decision is either the new subset of features should replace the old one or, ignore the new subset and keep the old selected subset of features. Subset generation and evaluation should be repeated until satisfying a specified stopping criterion [132]. Subset generation has two governing aspects; the first one is the starting point, which in turn affects the searching algorithm. Therefore, if the search for a subset of features starts with the empty set and then adds features while searching, this is called forward selection. Whereas starting with a full set of features then remove features during searching this is called backward selection. Moreover, starting with both sides and adds or removes features simultaneously this is called Bidirectional. The second governing aspect is the searching strategy. So far, three searching strategies are prevalent. The first strategy is

complete search, the second is sequential search, and the third is random search. Those three search strategies will be explained concisely in the following sections.

- ***Complete search***

  This search strategy aims to find the optimal result by doing a complete search. The optimal result can be determined by evaluation criterion. Andrew Ng mentioned that branch and bound algorithm are considered to be a complete search example [133] whereas Doak considered beam search algorithm to be an example of complete search [134].

- ***Sequential search***

  This is a fast and easy to implement searching algorithm where the optimal result is not the primary concern. This approach adds or removes features one at a time. Sequential forward selection, sequential backward elimination, and bidirectional selection are considered examples of this technique. Another replacement is to add or remove a number of features in the first step and add or remove a lower number of features in the second step.

- ***Random search***

  This algorithm starts by randomly selecting a subset of features, then goes in two different directions. The first direction is to follow a sequential search technique. The second direction is to generate the second subset in a completely random manner. In other words, we cannot follow an absolute rule in generating subsets of features, i.e. no indication that the coming generated subset will be shrunk or expanded compared to the previous subset. This algorithm sometimes is called the Las Vegas [135].

3.1.2 Subset Evaluation

Subset evaluation is the second in the steps of feature selection as shown in Figure 2.55. The generated subset of features needs to be evaluated to decide either selecting this generated subset or generate another subset of features. In general, there are two main

criteria to evaluate a generated subset. The first is the independent criterion and the second is the dependent criterion. These two criteria can be implemented by using one of several models. The first is the filter model, the second is the wrapper model, and the third model is the hybrid model. The filter model depends mainly on characteristics of data to select features without using any mining algorithm [136]. Whereas the wrapper model uses a predetermined mining algorithm and uses it to evaluate the selected subset of features [137]. The hybrid model aims to have the advantages of both models. The filter model is easier to implement than the wrapper model as wrapper model is computationally expensive.

3.1.2.1 Independent criterion

Independent criterion can be implemented by using the filter model and depends on characteristics of training data without using a specific or predetermined mining algorithm. In order to extract useful characteristics from the training data, it needs to measure specifications of data such as distance, information, dependency and consistency. Those four measures will be explained in the following sections.

- ***Distance measures***

  A feature A is selected over feature B if feature A has a better capability to classify data than feature B does. This is determined by the amount of information each feature has on the different classes. The distance measure is considered an acceptable method to determine the ability of selected feature can separate between the different classes. In other words, if selecting feature A will generate longer distance between classes than feature B does , feature A will be preferred and chosen over feature B.

- ***Information measures***

  Information measures quantify the amount of gained information from the training data after selecting feature A. Feature A is preferred over feature B if feature A can gain more information from the training data than feature B does.

- ***Dependency measures***

  It measures how much the selected feature is correlated or associated to a specific class. Assume that we have two selected features the first feature is called A and the second feature is called B. By measuring dependency, we conclude that feature A is highly correlated or associated to class C. In this case, the filter model will select feature A as it will be an improvement in classifying our data.

- ***Consistency measures***

  It measures how much the minimum selected features are consistent in classifying data in comparison with full features classification performance. It measures the consistency between minimum selected features in classifying training data and full features' performance in classifying the same training data. The selected features would be inconsistent if the data were more misclassified than implementing the full features method.

3.1.2.2 Dependent criterion

Dependent criterion uses the wrapper model. Thus, it uses a predetermined mining algorithm, which results in better results but uses more complex computational techniques than the filter model does and also, it is considered harder to implement than the independent criterion, which uses the filter model.

3.1.3 Stopping criterion

Stopping criterion is used to stop the selection algorithm depending on specific conditions. Once this condition is verified the feature selection algorithm should be stopped. As stopping condition may arise because the search of features completes, or because a maximum number of iterations has been reached, or a minimum number of features has been selected. Moreover, a stopping criterion may be reached because adding or deleting features has not improved the result or because the selected features cause a lower classification error than the allowable one. The allowable classification error varies from an application to another.

3.1.4 Result Validation

After features have been selected and the full selection criterion stopped, the results need to be validated. Classification error is a functional parameter to evaluate the performance to be either accepted or rejected. To accept the result, the classification error should be lower than the allowable one where allowable error is variable according to the application.

3.2 Implementation of feature selection for biosignals

In this section, the implementation of feature selection with high dimensional dataset will be illustrated. The model used sequential feature selection, which is a very popular feature selection algorithm. The performance of the selected features was evaluated by using holdout and cross-validation techniques.

Some types of data had a substantial number of features with a limited number of observations. A large number of features, in this case, did not lead to distinguished observations but might lead to overfitting to the noise. Moreover, reducing features saved computation time and decreased parameters related to observations.

There are two principal methods to diminish features: feature selection and feature transformation. Feature selection algorithms select a subset of features from the original feature set; on the other side, feature transformation methods map data from the original high-dimensional feature space to a new space of reduced dimension. The data were collected, by our colleague Dr Khairul Anam, using Flex comp Infiniti system for ten combined finger movements as shown in Figure 3.2.

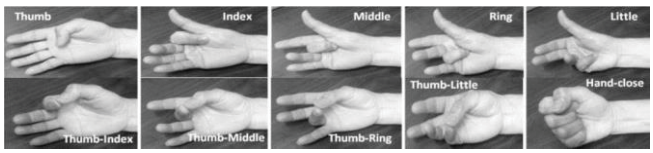


Figure 3. 2: Ten targeted finger movements [75]

The data set is divided into three sections. Two of them were dedicated to training while the third part was used for testing. The main set of features, before reduction, was 5000. The features were selected using the training data, and the evaluation of the performance of the selected features was done on the test data. This algorithm called holdout validation. Another simple and widely used method for evaluating and selecting a model is cross-validation method.

Filters are usually employed as a pre-processing step since they are fast and straightforward. A commonly used filter method for data was to apply a criterion where the P-value was computed for each feature as a measure of how useful it is at separating groups. The empirical cumulative distribution function (CDF) of the P values is plotted as shown in Figure 3.3.

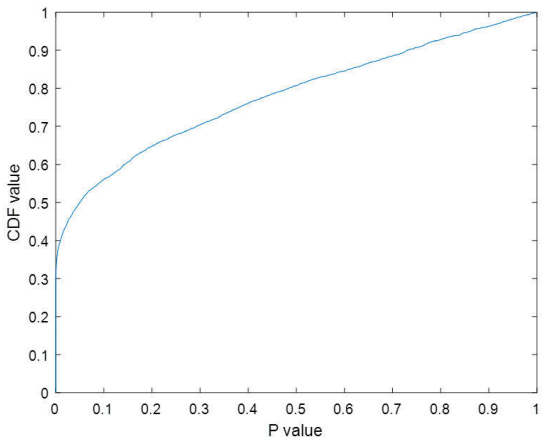


Figure 3. 3: CDF values versus P values

35% of features had P values close to zero and over 50% of features had P values smaller than 0.05, meaning that there were more than 2500 features among the original 5000 that had strong discrimination power. One could sort these features according to their p values and select some from the sorted list.

The most straightforward way to decide the number of needed features was to plot the MCE (misclassification error, i.e., the number of misclassified observations divided by the number of observations) on the test dataset as a function of the number of features.

For the data used in this example, the holdout partition and the sizes of two groups dictate that the most considerable allowable number of features for applying quadrature discriminant analysis (QDA) was about 70. Now, MCE was computed for various numbers of features between 5 and 70 and showed the curve of MCE as a function of the number of features. To reasonably estimate the performance of the selected model, it was essential to use the training samples to fit the QDA model and compute the MCE on the test observations (blue circular marks in Figure 3.4) and resubstitution MCE (red triangles in Figure 3.4).

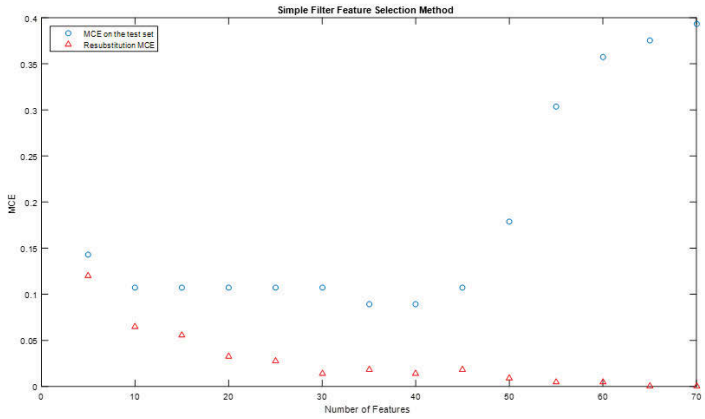


Figure 3. 4: Misclassification Error

The resubstitution MCE was over-optimistic. It consistently decreased when more features were used and dropped to zero when more than 60 features were utilised. However, if the test error increased while the resubstitution error decreased, then overfitting might occur. This simple filter feature selection method obtained the smallest MCE on the test set when 15 features were employed. The plot showed over-fitting began to occur when 20 or more features were used. The smallest MCE in the test set was 12.5%.

The filter feature selection procedure was used as a pre-processing step since it was fast. More advanced feature selection algorithms improved the performance. Sequential feature selection was one of the most broadly implemented techniques. It selected a subset of features by sequentially adding (forward search) or removing

(backward search) until certain stopping conditions were satisfied. The result of forward selection using cross-validation is shown in Figure 3.5.

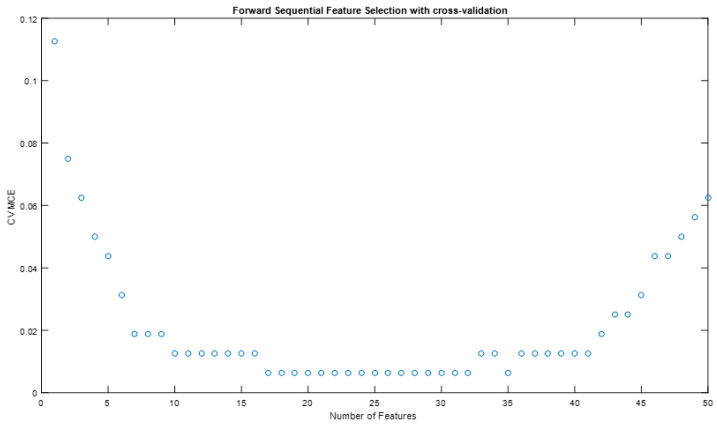


Figure 3. 5: Forward Sequential Feature Selection

The cross-validation MCE reached the minimum value when ten features were used and this curve stayed flat over the range from 10 features to 35 features. In addition, the curve went up when more than 35 features were implemented, which meant over-fitting occurred there. It was preferable to have fewer features, so here ten features should be suitable. Figure 3.6 shows that the resubstitution MCE values were overly optimistic. Most were smaller than the cross-validation MCE values, and the resubstitution MCE went to zero when 16 features were employed.

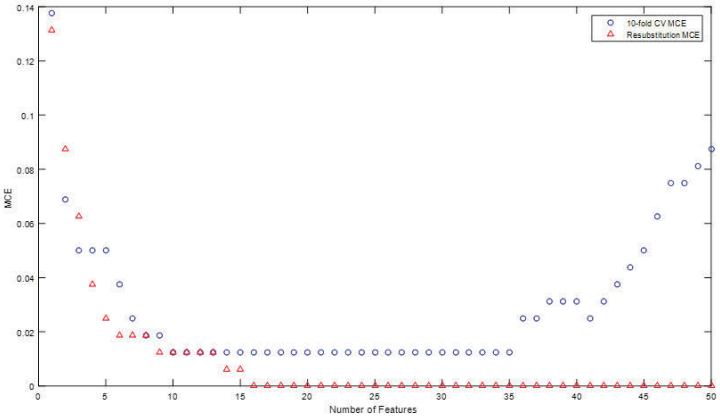


Figure 3. 6: MCE for resubstitution and cross folded (QDA)

If we replaced the classification methods, used in previous steps, by another technique like extreme learning machine, the error value was less than Quadratic Discriminant Analysis technique as shown in figure 3.7. The cross-validation MCE reached the minimum value when 12 features were used and this curve stayed flat over the range from 12 features to 32 features. Furthermore, the error increased when more than 32 features were utilised, which meant over-fitting occurs there.

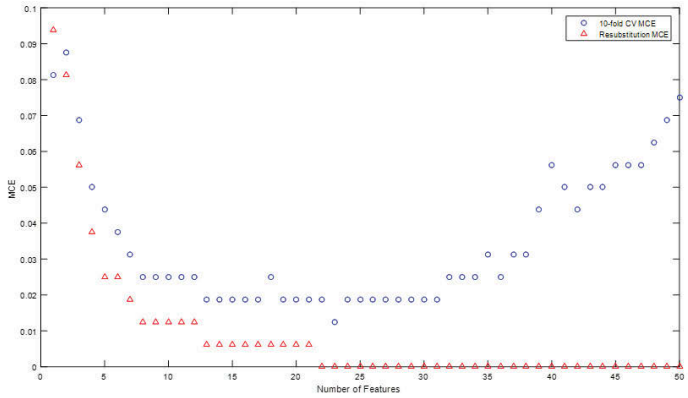


Figure 3. 7: MCE for resubstitution and cross-folded (ELM)

We concluded that the resubstitution error was not an adequate performance estimate for evaluating and selecting features. We wanted to avoid using resubstitution error, not only during the final evaluation step but also during the feature selection procedure. Cross-validation MCE was considered as an accurate measurement of the selected features performance.

3.3 Feature Learning

In machine learning, feature learning or representation learning is a set of techniques that learn features by itself via transforming data input to a representation that can be exploited efficiently in machine learning tasks. Feature learning obviates the need for manual feature engineering, which is otherwise necessary. In another word, feature learning is to train the proposed model to learn how to learn [138]. The performance of any machine-learning model immensely depends on how convenient is the representation or feature to reveal any variation in the input

data. The learnt set of representations are considered as efficient ones in case of being able to extract discriminative information from any input data and capture the descriptive factors from input data. The motivation behind feature learning was to save both time and effort that were spent in manually extracting features, as manual extracting features techniques are incredibly dependent on user's experience because each application has its own set of features that best extract discriminative information from input data. Features learning may be either supervised or non-supervised methodology. The supervised technique is due to the labelled data while the unsupervised one is because of unlabeled one. The supervised feature learning can be accomplished by supervised dictionary learning, multilayer perceptron and supervised neural networks. On the other side, the unsupervised feature learning can be executed by independent component analysis, autoencoder, matrix factorisation [139], many clustering techniques [140-142], and supervised dictionary learning. In the following section, a brief demonstration of different feature learning techniques will be illustrated.

3.3.1 Supervised learning

As it was mentioned above, the term supervised learning is derived from labelled data, which have, the classes or targets known in advance.

- ***Supervised dictionary learning***

  In this methodology, a set of features or representations were developed from the input data known as the dictionary. The established dictionary utilised each data point which was represented as a weighted sum of the dictionary representative elements. Both weights and dictionary representative elements were estimated via minimising the error across the input data with implementing sparsity whereby only a few number of data points have non-zero weights via hiring $L1$ regularisation approach. Therefore to improve the dictionary elements, further optimisation were applied by minimising the classification error through $L2$ regularisation algorithm [143].

- ***Neural networks***

  Neural networks are prevalent in the field of feature learning and have been widely used. In our research, we implemented different types of neural networks for the purpose of feature learning. The inputs to the neural network were the data while the output represented the features. The structure of the network consists of the typical network, as it has an input layer, hidden layer, and output layer. However, the methodology of estimating the output was entirely different, when compared to classifier networks. Later in this chapter, more details will be demonstrated on neural networks.

3.3.2 Unsupervised learning

The unsupervised learning is associated with an unlabeled dataset where the user does not know the corresponding class or target for each data point. In case of unlabeled data, the feature learning objective is to capture the underlying or hidden descriptive factors in the input data. The data are then labelled, as each feature has its corresponding class and the entire process is known as semi-supervised learning [144]. Various approaches utilised for unsupervised feature learning will be illustrated in the following section.

- ***K-means clustering***

  K-means clustering is a technique where the unlabelled data are divided into k clusters. Assume that we have a dataset with n vectors; each vector is mapped to a cluster based on estimating the mean value where each vector was mapped to the closest mean cluster. The feature generation can be easily estimated by applying several mathematical operations on the cluster. For example, but not limited to, the centroid of clusters is used to generate features as the clustering of the unlabeled data are mapped into different clusters. The geometric centre is computed for each cluster by adding k binary features on each sample. In which the i feature equals to 1 in case of having its

centroid closest to the i^{th} cluster [140]. Furthermore, the distance to the cluster can be used as a feature and processed via radial basis function [145]. Researchers in the field of feature and deep learning recognised that aspects of the k-means features perform in a similar behaviour to sparse coding techniques [146]. Coates et al. concluded that the appropriate transformation of k-means competes for the performance of advanced restricted Boltzmann machine (RBM) and autoencoder in the field of image processing and classification [140]. K-means showed a superior achievement in the area of Natural language processing [147].

- ***Principal component analysis***

  The principal component analysis was illustrated in chapter 2 in details. It is a well-known algorithm for dimensional reduction, where the dimension of unlabelled data is reduced linearly and transformed orthogonally to a new domain. PCA is adequate for highly correlated data. Whereas in the case of the uncorrelated data, the reduction leads to undescriptive factors as PCA presumes that the orientations of considerable variance are the orientations of the most interest, which may be not suitable in all situations. Furthermore, the original data set is transformed in an orthogonal direction. PCA takes advantage of the first and second moment only, which may not be satisfactory to generate descriptive factors from the data. The shortcomings of PCA implementation as feature learning impose limitations in its employment for feature learning purposes.

- ***Local linear embedding (LLE)***

  Local linear embedding is a methodology where the high dimensional data are reduced to a lower dimensional level while preserving the properties of the neighbourhood. LLE passes through two crucial stages to generate lower dimensional data. The earliest stage is called "neighbour-preserving" where the primary purpose is to maintain the geometric properties of the neighbourhood. At the neighbour-preserving stage, the data point input,x_i , is restored by adding the weights of the k nearest neighbours whereby

the optimum weights values are computed via minimising the reconstruction error. The reconstruction error is estimated by the difference between the original data point and the restored. The last stage in LLE is the dimension reduction whereby the previously calculated optimised weights are utilised in the dimension reduction step. Using the same optimised weights in both stages, ensures the maintenance of geometric properties of the neighbourhood [148]. In comparison to PCA, LLE showed superior performance than PCA in the area of feature learning.

- ***Independent component analysis (ICA)***

  Independent component analysis is an approach whereby the features are learnt or generated by the weighted summation of the independent components of non-Gaussian distribution of the original data [149]. The hypothesis of non-Gaussian will not be applicable in any Gaussian distribution dataset, as the weights cannot be estimated.

- ***Unsupervised dictionary learning***

  Sparse coding is a convenient example of unsupervised dictionary learning, as it does not use labelled data to establish the dictionary elements. However, in sparse coding, the features or data representations are implied from the unlabelled data. The dimension of dictionary elements is more abundant than the dimension of the unlabelled data [150]. M. Aharon et al. proposed sparse learning algorithms to learn the dictionary elements [151]. They applied a generalisation to K-means clustering technique to achieve a methodology known as K-singular value decomposition (K-SVD). K-SVRD was recruited to learn sparse representations from unlabeled data.

3.4 Deep Learning

Deep learning is a promising algorithm that is defined by cascaded hidden layers. The prime objective of those cascaded layers is to learn the model, how to extract and transform features [152]. Deep learning may be known as deep machine learning, hierarchical learning or deep structured learning. Deep learning can be used for unsupervised learning as a pattern analysis method for unlabelled data or for labelled one. Deep networks contain an input layer, an output layer and in between layers which utilises processing for the input by either linear or nonlinear transformation [153-157]. Machine learning is the broad family and deep learning is considered as one class of the family [152]. The most vital step in deep learning is representing data, especially for low dimensional data. Deep learning will replace the traditional handcrafted feature extraction techniques, which depends on the user experience and application. This is will be achieved by a self-learning model from input training representations. Deep learning builds a hierarchical feature extraction system where it can be recalled for either semi-supervised or unsupervised learning methods [158]. The neural network is considered as one of the most popular techniques that can be employed for the execution of deep learning. The neural network represents responses of the neuron, within the brain, towards any stimuli. Deep neural networks such as convolutional neural networks, deep belief networks, autoencoder and others are being implemented for executing deep learning in many fields. As for example but not limited to, image classification, audio and speech recognition, language processing and many biosignal. Image processing showed spectacular and promising results when implemented with deep learning techniques [159]. In brief, deep learning transforms the input representations through more profound and cascaded layers than the shallow algorithm where every layer has parameters such as neuron to be determined through training stage.

The history of deep learning started in 1965 where Ivakhnenko et al. mentioned that the multi-layer perceptron was hired to serve supervised learning [160]. Later in 1971, the researchers proposed a deep learning model of eight cascaded layers [161]. From this point, the neural

networks algorithm started to serve deep learning technique [162]. Another trial was created by LeCun et al. to apply backpropagation technique to recognise handwritten mail [163]. However, the simulation time was three days which is considered a long time relative to real-time applications. In 1993, other researchers applied a deep learning algorithm by using stacked recurrent neural networks. This proposal recalled a substantial number of subsequent layers, approximately 1200, which in turn consumed an extended simulation time. While other studies proposed two neural networks, the first one is to extract features, and the other one is for classification. In 1995 Carvalho et al. suggested implementing deep learning by using six layers with awake sleep algorithm [164, 165]. where the authors executed a deep learning system. However, it took about two days for the simulation. Many factors affected the long simulation time. One of those factors is the problem of vanishing gradient which was described in details in studies by Hochreiter et al. [166, 167].

Speech signal recognition by shallow or deep learning was studied for a long time [168-170]. However, none of the shallow or deep methods showed any improvement on Gaussian mixture Hidden Model [171]. The weakness that faced researchers during these days was vanishing gradient, the weak correlation between neural networks stages, the low capability of computers in these early days and low scale dimensional data as well [172, 173]. All these reasons led to avoid the hiring of deep neural networks for speech recognition purposes and preferring Gaussian mixture hidden model to it. Another useful technique for speech recognition is called long short-term memory (LSTM). This technique overcame vanishing gradient problem and could recall an event that happened from thousands of stages ago [174]. Later a technique called connectionist temporal classification (CTC) and stacked with long short-term memory (LSTM) started to emerge [175]. This algorithm is implemented in google speech recognition which is currently existing in smartphones.

The deep neural network is a ubiquitous technique used to execute deep learning concept as the deep neural network is a multi-hidden layer between input and output [176]. The traditional deep neural network is feedforward. However, both recurrent neural networks and long short-

term memory showed superior performance for language modelling field [177]. The method of training the network can also follow back propagation. The essentials of implementing backpropagation and train neural networks are mentioned in [178, 179]. The backpropagation was standardised in 1993 to be used internationally [180]. The weights of backpropagation will be estimated via gradient descent stochastic method as shown below:

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \eta \frac{\partial C}{\partial \omega_{ij}} + \xi(t) \tag{3.1}$$

Where η is the learning rate, C is the cost function, and $\xi(t)$ is the stochastic term.

Both cost and activation functions are strongly dependent on the type of learning either supervised or unsupervised. If we have supervised learning the common cost function will be cross entropy and softmax for activation function for multiclass. The softmax activation can be expressed as follows:

$$\rho_j = \frac{\exp(x_j)}{\sum_k exp(x_k)} \tag{3.2}$$

Where ρ_j is the probability of the output class for the j^{th} input, x_j is the j^{th} input, k is the total number of inputs and x_k is the k^{th} input.

Moreover, cross entropy can be expressed as follows:

$$C = -\sum_j d_j \log(\rho_j) \tag{3.3}$$

Where d_j is the targeted class and ρ_j is the actual output class [181].

The simulation time and overfitting are pervasive problems that may associate implementing neural networks in deep learning concept. Sparsity and weight decay are well-known techniques to reduce overfitting. A newer methodology called dropout is being applied to reduce overfitting. Whereby some hidden layers are omitted in a random manner [182].

Convolutional neural networks are considered one of the most widely used neural networks in deep learning applications for two-dimensional data and visual one [183]. The networks contain of concatenated layers of convolutional where maximum pooling and tied weights are applied [184].

Let us assume that we have the error function E. Where error concerning neuron is identified as $\frac{\partial E}{\partial y_{ij}^{l}}$ .The chain rule to be applied to estimate the gradient component for each weight. This is the contribution for each term.

$$\frac{\partial E}{\partial \omega_{ab}} = \sum_{i=0}^{N-m}\sum_{j=0}^{N-m}\frac{\partial E}{\partial x_{ij}^{l}}\frac{\partial x_{ij}^{l}}{\partial \omega_{ab}} = \sum_{i=0}^{N-m}\sum_{j=0}^{N-m}\frac{\partial E}{\partial x_{ij}^{l}} y_{(i+a)(j+b)}^{l-1} \tag{3.4}$$

where $\frac{\partial x_{ij}^{l}}{\partial \omega_{ab}} = y_{(i+a)(j+b)}^{l-1}$. We need to know $\frac{\partial E}{\partial x_{ij}^{l}}$ to estimate the gradient.

$$\frac{\partial E}{\partial x_{ij}^{l}} = \frac{\partial E}{\partial y_{ij}^{l}}\frac{\partial y_{ij}^{l}}{\partial x_{ij}^{l}} = \frac{\partial E}{\partial y_{ij}^{l}}\frac{\partial}{\partial x_{ij}^{l}}\left(\sigma\left(x_{ij}^{l}\right)\right) = \frac{\partial E}{\partial y_{ij}^{l}}\,\sigma'(x_{ij}^{l}) \tag{3.5}$$

Where $\frac{\partial E}{\partial y_{ij}^{l}}$ is the error of the current layer while the error of the previous layer can be estimated as follows:

$$\frac{\partial E}{\partial y_{ij}^{l-1}} = \sum_{a=0}^{m-1}\sum_{b=0}^{m-1}\frac{\partial E}{\partial x_{(i-a)(j-b)}^{l}}\frac{\partial x_{(i-a)(j-b)}^{l}}{\partial y_{ij}^{l-1}} = \sum_{a=0}^{m-1}\sum_{b=0}^{m-1}\frac{\partial E}{\partial x_{(i-a)(j-b)}^{l}}\omega_{ab} \tag{3.6}$$

The recursive neural networks are trained in a reverse method via differential algorithm [185] where weights are being estimated in a recursive manner [186]. Recursive neural networks are performed in natural language [187] and the tensor domain [188].

Deep belief networks are composed of multilayers which are trained following unsupervised learning, and each layer is created by restricted Boltzmann machine [189]. The typical restricted Boltzmann machine that has weights matrix $W = (\omega_{i,j})$ of size $(m * n)$, hidden units h_j, visible units ϑ_i, bias weights for hidden units b_j and bias weights for visible units a_i.

The energy for this configuration can be estimated as follows:

$$E(\vartheta, h) = -\sum_i a_i \vartheta_i - \sum_j b_j h_j - \sum_i \sum_j \vartheta_i \omega_{i,j} h_j \tag{3.7}$$

Alternatively, the matrix domain can be represented as follows:

$$E(\vartheta, h) = -a^T \vartheta - b^T h - \vartheta^T W h \tag{3.8}$$

Where the probability distribution function for this configuration can be expressed by using energy function.

$$P(\vartheta, h) = \frac{1}{Z} e^{-E(\vartheta,h)} \tag{3.9}$$

Where Z is a normalising constant to ensure that the sum of all probability values will be equal to one. It can be estimated by the sum of all values of $e^{-E(\vartheta,h)}$ that fit all possible configuration. The hidden and visible units' functions are mutually independent. Assuming that m represents the total number of hidden units and n is the total number of the visible units. The conditional probability of the configuration of ϑ visible units and h hidden units can be expressed as follows:

$$P(\vartheta|h) = \prod_{i=1}^{m} P(\vartheta_i|h) \tag{3.10}$$

Moreover, the conditional probability of h hidden units knowing ϑ visible units is given by

$$P(h|\vartheta) = \prod_{j=1}^{n} P(h_i\,|\vartheta)$$

(3.11)

So the individual probabilities can be expressed as

$$P\left(h_j = 1\middle|\vartheta\right) = \sigma(b_j + \sum_{i=1}^{m} \omega_{i,j}\vartheta_i)$$

(3.12)

$$P(\vartheta_i = 1|h) = \sigma(a_i + \sum_{j=1}^{n} \omega_{i,j}h_j)$$

(3.13)

Where σ is the logistic sigmoid function.

Automatic speech recognition is considered as a widely implemented deep learning application after hiring long short-term memory technique for recurrent neural networks [174]. Moreover, natural language was served by deep learning [190]. Also, deep learning showed a superior performance in image recognition where the researchers used MNIST dataset. They employed 60000 handwritten digits as a training set and 10000 as a testing set. The best-achieved results were verified in [191] of error 0.23%. Drugs and toxicology recognition was classified by deep learning [192]. The field of customer relationship management was classified by deep learning and showed a good performance [193]. Moreover, the recommendation systems implemented deep learning for extracting better significant features [194]. In the field of bioinformatics, the autoencoder was implemented as a deep learning to recognise gene ontology [195].

3.5 Artificial neural networks

Artificial neural network (ANN) is the artificial imitative of the natural brain. The neural networks consist of neurons, which are similar to axons in the natural brain. The neurons are linked by synapses, which are responsible for transmitting the signal between neurons. Figure 3.8 show a simplified structure of natural brain while Figure 3.9 demonstrates the neural network structure.

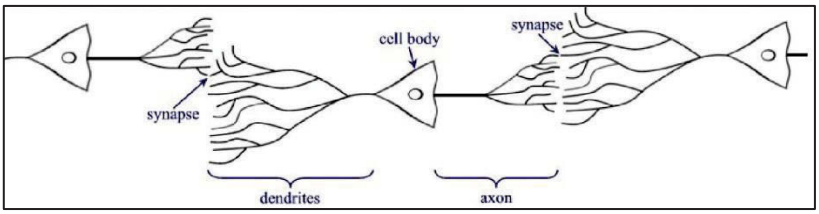


Figure 3. 8: The structure of natural brain [196]

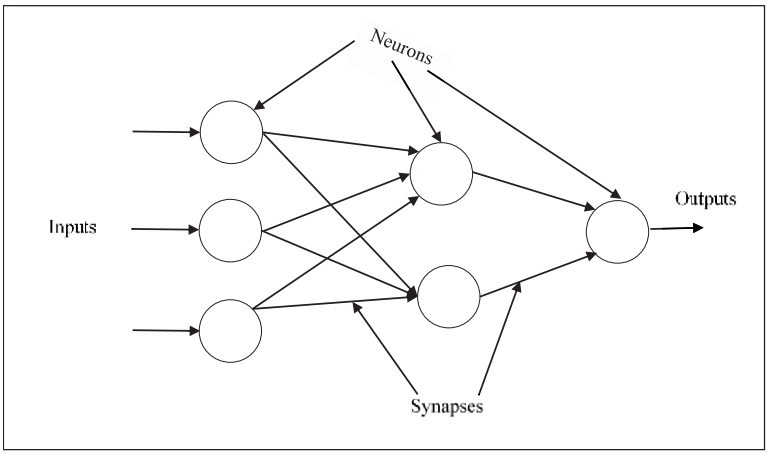


Figure 3. 9: The structure of neural networks

The idea of neural network appeared in 1943 when two researchers implemented threshold logic to serve the mathematical model of neural networks [197]. This model sparked the field of neural networks and motivated researchers develop two different approaches. The first approach was understanding the behaviour of the biological brain whereas the second method was mapping this behaviour in artificial neural networks [198]. In 1949, Hebb proposed an

unsupervised learning model which was later known as Hebbian learning algorithm [196]. Calculators are considered the first implementation of Hebbian networks into real applications [199]. In 1956, other researchers suggested different neural networks computation techniques [200]. The perceptron concept was first introduced in 1958 where it was mainly employed for pattern recognition. The perceptron has two layers that use basic mathematical operation as addition and subtraction [201] then backpropagation was presented by Werbos in [202]. In the same context, researchers found that the simple perceptron was not able to execute executive OR and the ability of the computers, within this period, was not capable of managing larger neural networks [203].

The artificial neuron is counted as a reduction to the natural neuron. Each artificial neuron has an input, an output and a cell body. Synapses establish the connection between either input and cell body or output and cell body. The artificial neuron multiplies several inputs by adaptive weights then integrates them to the output.

Assume that we have a neuron j receives an input $P_j(t)$ from the antecedent neurons; this neuron contains four main functions [204].

- An activation function $a_j(t)$. The activation function can be linear, ramp, sigmoid, Gaussian or others [205].
- A threshold function θ_j that can be adapted based on the learning function.
- A new activation function f at certain time $t+1$ that can be estimated via the combination of activation function $a_j(t)$, threshold function θ_j as well as input $P_j(t)$

$$a_j(t+1) = f(a_j(t), P_j(t), \theta_j) \tag{3.14}$$

- The output function $o_j(t) = f_{out}(a_j(t))$ that computes the output from activating the artificial neuron

The connection between consecutive neurons can be managed by applying weights. Assume that we have two successive neurons, where i is the ancestor neuron while j is the inheritor one. The connection between i and j is allocated a weight w_{ij}.

The propagation function counts the input of a neuron $P_j(t)$ by recruiting both the output of the antecedent neuron $o_i(t)$ along with the weights w_{ij}.

$$P_j(t) = \sum_i o_i(t) w_{ij} \tag{3.15}$$

The learning task is considered as the most critical stage in neural networks. A cost function is assigned to the network whereby the primary objective of this cost function is to find a solution. This should be close to the optimum solution and charge the lowest cost function over the data.

3.5.1 Neural networks architecture

The artificial neurons can be connected in many forms to establish the different architectures of the network. Different network architectures will now be demonstrated.

- ***Single layer feedforward networks***

  Single layer feedforward network is considered a modest architecture as it has only an input layer and output layer. The input layer has no reckoning as it evades the inputs to the output without any further processing. Figure 3.10 displays the single layer network architecture.

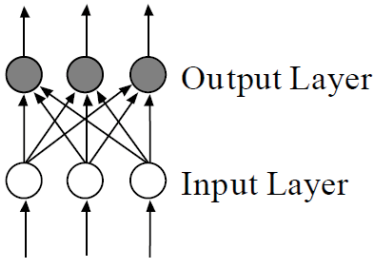


Figure 3. 10: The single layer feedforward networks architecture [206]

- ***Multilayer feedforward networks***

This architecture is more complicated than the single feedforward idea as it has single or multi-hidden layers that contain multiple numbers of neurons. As it is clear that this network can be implemented to solve advanced problems rather than the single approach [206]. The multi-layer network may be fully connected or not fully connected. In the fully connected network, every neuron is connected to each other of the next layer, while in the case of not fully connected, not all neurons are joined to the neurons of the subsequent layer. Figure 3.11 and figure 3.12 show the fully connected and not fully connected network architecture respectively.

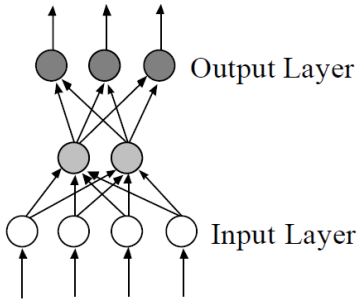


Figure 3. 11: The multi-layer fully connected feedforward networks architecture [206]

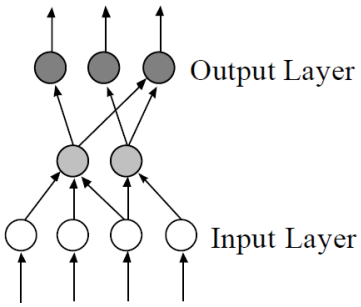


Figure 3. 12: The multi-layer not fully connected feedforward networks architecture [206]

- ***Recurrent networks***

The recurrent network has at least one feedback loop from the output of the neuron to the input of another neuron. The feedback may exist between neurons in the same layer or between neurons of different layers. Furthermore, the feedback may occur between the same neuron as the output is returned to the input of the same neuron. Figure 3.13 illustrates the architecture of recurrent artificial neural networks.

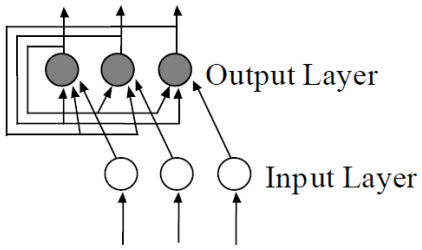


Figure 3. 13: The recurrent network's architecture [206]

- ***Lattice networks***

The lattice network is formed by an array of neurons whereby no limitations can be applied to the dimensions of the array. The input neuron should be connected to all the neurons of the array. Figure 3.14 demonstrates the architecture of the lattice networks with three by three network.

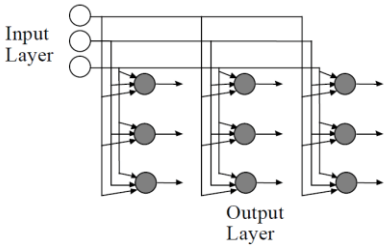


Figure 3. 14: The lattice networks architecture [206]

3.5.2 Neural networks learning techniques

The learning techniques are the approaches which are used to learn and estimate the values of the network weights that best match the application. It provides results with

lowest cost function. Three prime methodologies are used to learn weights, which are supervised learning, unsupervised learning and reinforcement learning. Some brief demonstration will be illustrated on the three approaches in the following section.

- ***Supervised learning***

  In the supervised learning approach, the classes or targets of data are known beforehand. The objective of the learning technique is to find the best function for mapping the inputs with lowest cost function [207]. Assuming that we have an input data x where $x \in X$ and equivalent are known in advance classes y where $y \in Y$ the main purpose of the learning is to find the function f that leads to $f: X \rightarrow Y$ with the lowest cost function. One of the widely used cost functions is the mean square error one. This function estimates and reduces the error between network output $f(x)$ and targeted classes y to its lowest values. The cost function can be reduced via various techniques including gradient descent to find the local minimum of the function. The gradient descent can be accomplished via multilayer perceptron following backpropagation procedure. Speech recognition, pattern recognition, handwritten letters and motion recognition are all examples of supervised learning.

- ***Unsupervised learning***

  In the unsupervised methodology, the targeted classes are unknown. With the assumption that we have a data x while cost function is a function of x and output of the network f. The cost function to be reduced to its lowest values following various techniques. where it is strictly dependent on the data input. The cost function can be any function including the mean square error. Assume that we have data input x and the network model is represented by $f(x) = a$ where a is a constant the cost function can be estimated via mean square error $C = E[(x - f(x))^2]$ and to be reduced to its minimum value. Another cost function like posterior probability and mutual information between x and $f(x)$ however, these functions should be increased to its

maximum values and minimised. Clustering and statistical distributions are examples of unsupervised learning.

- ***Reinforcement learning***

  Reinforcement learning is considered the most challenging approach as it does not have a data input. However, observations, targets and cost function are generated by cooperation between a mediator and the surrounding environment. At each time t, the observation x_t, target y_t and cost function c_t are generated while the objective is to minimise the cost function. The environment is modelled as Markov decision process (MDP) that has states $s_1, s_{2,\dots\dots,} s_n \in S$ and actions $a_1, a_{2,\dots\dots,} a_m \in A$. The interaction between model and a mediator results in generating three probability distribution functions. The first produced probability distribution is the observation distribution $P(X_t|S_t)$ while the second is the cost distribution $P(C_t|S_t)$ and the third is the transition between states distribution $P(s_{t+1}|s_t, a_t)$. The corporation between those three distributions establishes a Markov chain (MC) that minimise the cost function. Artificial neural networks are always used with the reinforcement learning [208, 209]. Moreover, ANN with dynamic programming [210] is important in solving multilinear problems for different fields [211-214]. Nearly all sequential decision-making problem utilises reinforcement learning.

Following the types of artificial neural networks that were mentioned above, some of the widely applied networks will be illustrated concisely in the following section

- ***Autoencoder***

  Autoencoder is a widely used feed-forward artificial neural network. However, the output has the same number of input, as its objective is to reconstruct the input at the output. Autoencoder is known as Diabolo network or autoassociator [153, 215]. The autoencoder contains an input layer, output layer and one or more hidden layers, using a similar architecture to the multilayer perceptron (MLP). The target from autoencoder

is to reconstruct the input at the output, not like MLP, therefore it follows the unsupervised learning algorithm. In our research, we implemented autoencoder as a feature learning layer to learn features from surface electromyography signals for ten-finger movements [216]. More details will be demonstrated on this study in chapter 4. The autoencoder was utilised in many different applications as for learning generative models by Kingma et al. [217, 218]. And for dimension reduction alongside efficient coding [219, 220].

- ***Probabilistic neural network***

The probabilistic neural network (PNN) is another example of feedforward neural network with four layers. It has the input, output and hidden layer in addition to a summation or pattern layer. The probability density function for each class is estimated via non-parametric algorithm along with Bayes approach to assigning a class with the highest posterior probability [221]. As PNN employs the Bayesian procedure, it is deduced from the Bayesian network [222] and a kernel model with linear discriminant analysis [223]. The major fields of applications are both pattern recognition and classification.

- ***Time delay neural network***

Time delay neural network (TDNN) is a typical example of feedforward networks that use continuous data as an input to learn features from the input stream of data. A time shift or delay is encountered on every data sample to allow analysing multiple data points at the same time and learn features from them. TDNN connections weights are trained via backpropagation like perceptron networks. TDNN uses supervised learning for pattern recognition applications.

- ***Convolutional neural networks***

Convolutional neural network (CNN) is a category of feedforward network that is applied for image processing. Convolutional neural networks are mainly driven by the animal visual cortex [224] whereby an only a small amount of neurons respond to the stimuli. Due to this convolutional neural network is one of the least pre-processing networks in comparison with other image processing and classification algorithms [225]. The convolutional network consists of an input layer, numerous hidden layers and output layer. The hidden layers are pooling, convolutional, normalisation and others. The convolutional neural network provides the opportunity to learn features from input images with less number of parameters than the other fully connected feedforward neural networks. It examines the tiling region size of the image rather than the whole image size [226]. Primary fields of applications are video and image recognition [194] and [215] respectively.

- ***Radial basis function networks***

Radial basis network (RBN) is a network that uses radial distance in referring to the centre. The radial basis function replaces the sigmoid activation function in the hidden layer for multi-layer perceptron. The activation function in RBN is usually a Gaussian one. Radial basis network has three layers, the input, the hidden and the output layer. The input layer is for transforming the inputs to the domain of the hidden layer. Whereas the hidden layer implements nonlinear activation function. And the output utilising the linear cooperation between hidden layer values to anticipate the output mean value in case of regression problem [145, 227]. However, in case of classification problem, the output activation function is a sigmoid one with a linear amalgamation for the posterior probabilities of hidden layer values. The radial basis network is similar to multi-layer perceptron as both avert looking for local minima because both networks

have one master concern, which is mapping the hidden layer to the output one. In radial basis network, it is preferred to link each data point to its centre, which in return guarantees a perfect coverage of radial basis function to the input domain. While it may cause overfitting that can be overcome by applying shrinking methods.

The input can be expressed as $x \in R^n$ while the output of network can be estimated as following

$$\varphi(x) = \sum_{i=1}^{N} a_i \, \rho(||x - c_i||)$$

(3.16)

Where N represents the number of neurons in the hidden layer, a_i signifies the weight for the i^{th} neuron in the output layer and c_i denotes the centre vector for the i^{th} neuron. The radial basis function is commonly computed as Gaussian.

$$\rho(||x - c_i||) = \exp[-\beta||x - c_i||^2]$$

(3.17)

Where the Gaussian is local to each centre vector as follows:

$$\lim_{||x|| \to \infty} \rho(||x - c_i||) = 0$$

(3.18)

- ***Self-organising map***

Self-organising maps (SOM) are the conventional back propagation recurrent neural networks. SOM is trained by using unsupervised learning technique to generate lower dimension data that make it very suitable for dimension reduction applications. The unsupervised concept arises from presenting any input to the network which does not require associating this input to its equivalent output. Self-organising map follows competitive learning, and not error correcting algorithm. Where all the neurons compete to respond to a subset of the input where only one neuron will activate and

will be the winner neuron among the grid of neurons. Sometimes the self-organising map is called Kohonen in relating to the professor Teuvo Kohonen who was the first to introduce SOM [228-230]. The self-organising map usually operates in two patterns either training or mapping. The training model generates the map based on input data while the mapping is to transform input data into the relevant class. Moreover, in the training pattern, the distance between vectors in high dimensional space is kept close to the lower dimensional space. This topology in self-organising map networks are known to preserve its topological properties for the original data. We implemented self-organising map techniques as deep learning to learn features from the input bio signal to classify ten finger movements [231].

3.6 Signal representation

Biosignals can be called any signals that were collected from any medical source or living organism. The collection of biosignal can be done at the tissue scale, cell scale or molecular scale. The biosignal normally records the electrical activity of an organ or for muscles. Electroencephalogram (EEG) represents the electrical activity of the brain while the electrocardiogram (ECG) represents electrical activity of heart and electromyogram (EMG) represents electrical activity of any muscle. The biosignal can be recorded for many reasons. It can be collected for diagnosis or analysing the muscles behaviour. However, biosignals can be considered as other signals that show high fidelity. Moreover, biosignals always follows the recruitment of a multichannel, which in turn carries more challenges with different signal processing techniques. The main purpose of signal processing and representation is to expel any noise that may be superimposed on the main signal and represent the signal in a manner that reveals any feature that might be hidden in the signal respectively. Noise exists on biosignals by default from the surrounding environment or from any device involved in collecting these signals. Power cables generate electromagnetic interference are also considered as noise. In general, Biosignal can be either continuous or discrete. Each of this type can be either predictable, random, fractal or messy.

The predictable signal can be expressed in the form of the biosignal with additive white Gaussian noise as mentioned in equation (3.19)

$$x(t) = s(t) + n(t) \tag{3.19}$$

A good example of predictable type biosignal is electrocardiogram or electromyogram. The main purpose of this predictable bio signal is to extract features that help in classifying each case and correlating between the biosignal and pathological cause.

Both electroencephalogram and R-R interval for electrocardiogram can be considered as good examples for random biosignals. The absence of its morphology mainly characterises random biosignal apart from the predictable signals. The indiscriminate or chaotic biosignal can be either static, which means there are no variations versus time or non-stationary due to its variations with time. The fluctuations with time may be indicative of a particular event. This event can be speculated by extracting features. Heart rate is an excellent example of the fractal signal, as the signal will not change by applying magnification or scaling techniques. The messy or chaotic signal is the most challenging signal to be predicted as its variations follow a nonlinear model. Subsequently, the chaotic signal can be resembled by a random signal that is being generated by a time-variant model.

The biosignal can be divided into a stationary and non-stationary signal. Each of this category has its analysis technique, which will be demonstrated in the following section.

Dividing the stationary bio signal into different frequencies is a common analysing algorithm. This algorithm can be employed by either fast Fourier transforms, periodogram method or parametric approach. In the following section, some of the essential signal representations algorithms will be presented. The primary signal representation approaches that will be displayed are spectrogram, wavelet and wavelet packet.

- ***Spectrogram***

The spectrogram is the representation of the spectrum frequencies representing certain biosignal. Numerically, spectrogram can be measured by calculating the square of the magnitude of Short-Time Fourier Transform (STFT). It is called short-term Fourier transforms rather than spectrogram. In short time Fourier transforms the long-time signal is divided into equal length segments and shorter in time. Short time Fourier transforms is relevant to Fourier transform. Then, the frequency and phase for each segment needs to be estimated separately. Based on the above, we can conclude that spectrogram can be considered as Fourier transform but for shorter segments rather than from the entire signal [232].

Assume that we have a discrete time signal x with a finite duration (limited signal) and a number of samples$N$. The Discrete Fourier Transform (DFT) can be expressed as follows:

$$\hat{x}(k) = \sum_{n=0}^{N-1} x(n) e^{-i\frac{2\pi k}{N}n} \quad , k = 0, \dots\dots, N-1 \tag{3.20}$$

Knowing that the Fourier transform is estimated at frequency $f = \frac{k}{N}$

The original signal x can be restored back to $\hat{x}$ by applying the inverse Discrete Fourier Transform as follows:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} \hat{x}(k) e^{-i\frac{2\pi k}{N}n}, \qquad n = 0, \dots\dots, N-1 \tag{3.21}$$

The two equations mentioned above can be rephrased as follows:

$$x = \frac{1}{N} F\hat{x} \tag{3.22}$$

$$\hat{x} = \bar{F}x \tag{3.23}$$

Where F is Fourier matrix of $n * n$ dimensions and $\bar{F}$ is its complex conjugate

$$F = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & e^{i\frac{2\pi}{N}} & e^{i\frac{4\pi}{N}} & \cdots & e^{i\frac{2\pi(N-1)}{N}} \\ 1 & e^{i\frac{4\pi}{N}} & e^{i\frac{8\pi}{N}} & \cdots & e^{i\frac{2\pi 2(N-1)}{N}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{i\frac{2\pi(N-1)}{N}} & e^{i\frac{2\pi 2(N-1)}{N}} & \cdots & e^{i\frac{2\pi(N-1)^2}{N}} \end{bmatrix}$$

Where the entrees of $\hat{x}$ are expressed in terms frequencies coefficients

$$f = 0, {}^{1}/_{N}, {}^{2}/_{N}, \ldots\ldots {}^{(N-1)}/_{N}$$

(3.24)

We need to calculate the spectrogram of the signal. Assume that we have a signal x of length N which is divided into successive equal segments m where $m < N$. The matrix of successive equal segments can be expressed as X where $X \in R^{mx(N-m+1)}$. The first column of X matrix equals $[x[0], x[1], \ldots\ldots x[m-1]]^T$ and its second column equals $[x[1], x[2], \ldots\ldots x[m]]^T$. The spectrogram for a signal x with window size m can be annotated $\hat{X}$. The columns which are composing matrix $\hat{X}$ is the discrete Fourier transform

$$\hat{X} = \bar{F}X$$

(3.25)

$$X = \frac{1}{m}F\hat{X}$$

(3.26)

The rows of the matrix $\hat{X}$ are representing the signal x in the time domain while its columns are representing the signal x in the frequency domain. Thus a simply spectrogram is a time-frequency representation for the signal x.

The spectrogram was used in many applications especially for speech signal analysis where Mergu et al. represented the speech signal by different representations for example, Fourier and spectrogram to conclude that the resolution is mainly dependent on used representations [233]. Some researchers estimated the time corrected version of rapid frequency spectrogram of the speech signal that showed a better ability to track the variations in the signal than other published techniques[234].

- ***Wavelet***

Wavelet is estimated by shifting and scaling small segmentations of the biosignal. Fourier transform is a representation of the signal as a sinusoidal wave with different frequency while wavelet is representing the abrupt changes of the signal. Fourier transform is considered as a better representation of the signal in the case of having smooth signals, while wavelet is believed to be a better representation, than Fourier, for the sudden changing signal. Since wavelet provides the opportunity to represent rapid variations of the signal and hence, help the system extract more discriminative features. So in brief wavelet is an analysis for time series signal that has non-stationary power at many frequencies [235]. Assume that we have a time series signal x_n with equal time spacing δt and $n = 0, \dots \dots N-1$ where the wavelet function is $\Psi_o(\eta)$ that depends on time η. The wavelet signal has zero mean and is represented in both time and frequency domain [236]. Morlet wavelet can be estimated by modulating our time domain signal by Gaussian as follows:

$$\Psi_o(\eta) = \pi^{\frac{-1}{4}} e^{i\eta\omega_o} e^{\frac{-\eta^2}{2}} \tag{3.27}$$

Where ω_o is the frequency of the unmodulated signal. The continuous wavelet of a discrete signal x_n is convolutional of x_n with scaled and shifted version of $\Psi_o(\eta)$

$$W_n(s) = \sum_{n'=0}^{N-1} x_{n'} \ \Psi^* \left[\frac{(n'-n)\delta t}{s}\right] \tag{3.28}$$

Where * is the complex conjugate and s is the scale.

The wavelet transform was applied in several studies and different fields. as Lau et al. implemented wavelet in Geophysics field [237]. where wavelet implemented for climate [238, 239]. And for weather [240, 241]. The above equation can be simplified by reducing the number of N. The convolutional theorem permits to estimate N

convolutional in Fourier domain by implementing Discrete Fourier Transform (DFT). The Discrete Fourier Transform for x_n.

$$\hat{x}_k = \frac{1}{N}\sum_{n=0}^{N-1} x_n e^{-2\pi ikn/N} \tag{3.29}$$

Where $k = 0, \ldots, N-1$ which is representing the frequencies. For a continuous signal $\Psi(t/s)$ is defined as $\hat{\psi}(s(\omega)$. Based on the convolutional theorem, the inverse Fourier transform is equal to wavelet transform as follows:

$$\psi_n(s) = \sum_{k=0}^{N-1} \hat{x}_k \hat{\psi} * (s\omega_k) e^{i\omega_k n\delta t} \tag{3.30}$$

Where the angular frequency ω_k can be expressed as follows:

$$\omega_k = \begin{cases} \frac{2\pi k}{N\delta t} & : k \leq \frac{N}{2} \end{cases} \tag{3.31}$$

or

$$\omega_k = \begin{cases} \frac{-2\pi k}{N\delta t} & : k > \frac{N}{2} \end{cases} \tag{3.32}$$

An improved copy of wavelet algorithm was recalled in [242] where the authors presented two techniques. The first method used expansion factors for filtering while the other approach was factoring in wavelet transform. Researchers introduced the Morlet wavelet to vibration signal of a machine [243]. The vibration signal of the low signal to noise ratio was represented by wavelet to grant fidelity to the signal and allow extraction better characteristic features. This model was implemented in [244] where researchers used wavelet transform to predict early malfunction symptoms that may happen in the gearbox.

- ***Wavelet packet***

The signal can be represented in both time and frequency domain simultaneously. This representation gains a fidelity to the signal due to its robust representation. Wavelet

packet is one of the very widely used signal representation that produces the signal in both time and frequency domain [245]. The wavelet packet shows a very well acted for both nonstationary and transient signals [246-248]. Linear combination of wavelets estimates wavelet packet. The coefficients of linear combination are calculated by recursive algorithm [249]. The wavelet packet estimation can be done as follows:

Assume that we have two wavelets type signal $h(n)$, $g(n)$ and two filters of length $2N$. Let us assume that the following sequence of functions is representing wavelet functions.

$$(W_n(x), n = 0,1,2, \ldots\ldots)$$

(3.33)

$$W_{2n}(x) = \sqrt{2} \sum_{k=0}^{2N-1} h(k) W_n(2x - k)$$

(3.34)

$$W_{2n+1}(x) = \sqrt{2} \sum_{k=0}^{2N-1} g(k) W_n(2x - k)$$

(3.35)

Where $W_0(x) = \varphi(x)$ is the scaling function, $W_1(x) = \omega(x)$ is the wavelet function.

$$N = 1, h(0) = h(1) = \frac{1}{\sqrt{2}} \qquad \text{and} \qquad g(0) = -g(1) = \frac{1}{\sqrt{2}}$$

(3.36)

The above equations became

$$W_{2n}(x) = W_n(2x) + W_n(2x - 1)$$

(3.37)

And

$$W_{2n+1}(x) = W_n(2x) + W_n(2x-1)$$

(3.38)

Where $W_0(x) = \varphi(x)$ is the scaling function, $W_1(x) = \omega(x)$ is the Haar wavelet function.

Many researchers implemented wavelet packet as Han et al. used wavelet packet to create an index called rate index to detect the damage that may happen to the structure of any beam [250]. In the same context, Wu et al. employed wavelet packet and neural networks to detect the fault in the combustion engine [251].

3.7 Data Augmentation

Data Augmentation is a technique that robustly invades the field of data mining and processing for regression and classifications purposes. Data augmentation denotes the techniques that may be used to generate iterative samples. The created latent samples are introduced to the original data to produce a high dimensional set. The new generated augmented data is used in training for the suggested model. Data augmentation algorithms become numerous and manipulation between them to obtain high accuracy results and at the same time, implementing the modest and rapid algorithm is an ordinary matter of talent. The proper selected data augmentation technique drives the accuracy values to a highly dramatic level. Researchers developed an approach to combine, search and select the best augmentation scheme between deterministic, marginal and conditional different augmentations methodologies. This approach is applied to three different classes of systems to attain the most efficient for sampling technique [252].

Augmentation may be applied in two domains; the first domain is data domain while the second is feature domain [253]. Many types of research demonstrated the art of data augmentation by generating numerous training samples [254-257]. Other researchers focused on the advantage of data augmentation and how it might act as an organiser to prevent overfitting associated during training of neural networks [258, 259] and develop the execution to avoid problems that may be correlated with the classes that are not represented equally [260].

Researchers in the field of data augmentation achieve higher accuracy values and enhance the performance of the classifier. A punch of distorted and warped samples of characters are generated by data augmentation technique [261]. This was not the only example of creating deformed samples of characters [262]. Where the malformed samples are generated randomly. This methodology was extended to be applied to backpropagation neural networks and reduced error rate to 0.4% on MNIST database [258]. Later, the researchers followed two augmentations techniques [253]. The first applied augmentation technique was data wrapping or data augmentation on input MNIST dataset before being introduced to convolutional neural networks. The output features from neural networks were augmented in feature domain. They employed support vector machine, extreme learning machine and backpropagation neural networks as classifiers whereby the error percentage ranged from 2.25% to 0% for training samples.

In the same context, data augmentation was hired by generating virtual samples [263]. Generally speaking, the virtual samples can be generated by the two techniques. The first methodology depends on generating virtual samples from relevant information. For example, in the field of image processing and recognition, we can generate virtual samples from the same image by producing a 3D view which in turn helps in creating virtual samples for the same image from different angles [264-268].

Let us assume that we have a dataset $e = (x, f(x))$ where e represents the original training samples. Where $x \in R_n$ and $f(x) = \{-1, 1\}$. Assuming that we have a previous information k, and we need to map our training set to the new domain. So, if we have a convert equals $T = y_T$, the data set e will be transformed via this conversion equation to generate virtual samples $(Tx, y_T(f(x))$. The generation of mathematical transformation T and y_T is mainly depended on the previous information which may result in either simple or complex transformation formula while the second algorithm depends on adding noise to the original signal [269].

Most of the techniques that were used to create virtual samples suffer a lack of combining reasonableness and adaptability simultaneously. Accordingly, we followed an algorithm to generate virtual Gaussian samples. This method calculated the mean and standard deviation for Gaussian distribution. Then the virtual samples could be generated following this technique, and finally, the virtual generated sample was added to the original set [270]. $\{x_1, \dots \dots . x_n, x_{n+1}, \dots \dots . x_k\}7$ represents the original data set belongs to R. The first n samples of the data set is continuous while the $k - n$ samples are discrete. A m random variables are generated by Gaussian algorithm $N = (\mu, \eta^2)$ for the first n consecutive samples knowing that μ represents the mean and η represents the standard error. However, for the samples $k - n$ that are assigned to be discrete ones, the values will not be transformed, and to keep the consistency between the discrete and continuous part, we generated random variables for the discrete part by using $N = (\mu, \eta^2)$ with η^2 equals zero.

3.8 Tensor

A tensor can be defined as a multidimensional array concerning a basis; however, for a vector, it can be represented as a single dimensional array for the same basis. In brief, tensors can be evaluated as a multi-dimensional vector. Tensors can be deemed as a mathematical method to represent values in a multi-dimension matrix. They are considered the complete version of a matrix, vector and scalar. Therefore, matrix, vector and scalar can be measured as sub-components of a tensor. The generation of tensor can be done by following transformation laws. Tensors are characteristics by having various coordinates systems. So the coordinates systems with its transformation laws will be analysed in the following section.

Assume that we have x^i where $i = 1,2, \dots \dots, N$ So by substituting the different values of i we can get N values of x in a N dimensional space $x^1, x^2, \dots \dots, x^N$. Moreover for the set of $\bar{x}^i$ can be expressed as $\bar{x}^1, \bar{x}^2, \dots \dots \dots, \bar{x}^N$ for N dimensional coordinates. In the same context, keeping the same transformation laws for x' to $\bar{x}'$ leads to the following transformation equation.

$$x^i = x^i(\bar{x}^1, \bar{x}^2, \ldots \bar{x}^N), \qquad i = 1,2, \ldots \ldots, N$$

(3.39)

The above equation creates an independent relation between the two different coordinates x^i and $\bar{x}^i$ for $i = 1,2, \ldots\ldots, N$. As long as the relation is kept independent, it can be recalled for transformation.

The Jacobin first order partial transformation will be estimated as follows:

$$J\left(\frac{x}{\bar{x}}\right) = J\left(\frac{x^1, x^2, \ldots\ldots, x^N}{\bar{x}^1, \bar{x}^2, \ldots, \bar{x}^N}\right) = \begin{vmatrix} \frac{\partial x^1}{\partial \bar{x}^1} & \frac{\partial x^1}{\partial \bar{x}^2} & \cdots & \frac{\partial x^1}{\partial \bar{x}^N} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial x^N}{\partial \bar{x}^1} & \frac{\partial x^N}{\partial \bar{x}^2} & \cdots & \frac{\partial x^N}{\partial \bar{x}^N} \end{vmatrix}$$

(3.40)

With an inverse transformation

$$\bar{x}^i = \bar{x}^i(x^1, x^2, \ldots\ldots, x^N), \qquad i = 1,2, \ldots \ldots, N$$

(3.41)

In brief both equations (3.39) & (3.41) can be expressed in the notation formula as follows:

$$x^i = x^i(\bar{x}), \qquad i = 1,2, \ldots \ldots, N$$

(3.42)

$$\bar{x}^i = \bar{x}^i(x), \qquad i = 1,2, \ldots \ldots, N$$

(3.43)

$\bar{x}$ can be concluded from x and $\bar{\bar{x}}$ can be deduced from $\bar{x}$ by recalling transformation. Assume that $\bar{x} = y$ and $\bar{\bar{x}} = z$. The transformations are represented by T_1, T_2 and T_3 where

$$T_1: \quad y^i = y^i(x^1, x^2, \ldots\ldots, x^N) \quad i = 1,2, \ldots\ldots, N \qquad \text{or} \qquad T_1 x = y$$

$$T_2: \quad z^i = z^i(y^1, y^2, \ldots\ldots, y^N) \quad i = 1,2, \ldots\ldots, N \qquad \text{or} \qquad T_2 y = z$$

T_3 can be deduced by the product of both T_1 and T_2

$$T_3: \quad z^i = z^i(y^1(x^1, x^2, \ldots\ldots, x^N), \ldots\ldots, y^N(x^1, x^2, \ldots\ldots, x^N)) \quad i = 1,2, \ldots\ldots, N \quad \text{or}$$

$$T_3 x = T_2 T_1 x = z \qquad \text{by considering} \quad T_3 = T_2 T_1$$

Where T_1, T_2 and T_3 represent the first, second and third coordinates transformations respectively.

3.9 Classifier fusion

In our research, we applied many optimisation techniques that were capable of improving the results. One of the implemented algorithms was classifier fusion. The results for our implemented systems will be discussed in chapters 4, 5, 6 and 7. However, in this section, a brief discussion will be presented on classifier fusion. Various classifier fusion approaches were developed that led to a substantial enhancement in the classification results. One of these techniques is the classifier fusion. Fusion concept can be applied to three levels either data level and called data fusion, feature level and named feature fusion or classification level and termed classifier fusion. The first two fusion techniques are beyond our scope while the classifier fusion will be illustrated in the following section. Classifier fusion has two approaches. The first approach works on classifier itself and keeps improving classifier performance by searching for the best combination of classifier sets once the best performance found the results could be released. Therefore, this method is used to select the best combination of classes and no results had been released before finding the targeted mixture [271-273]. The second scheme depends on the classifier output and not the classifier itself as the foregoing technique [274-277]. Dynamic classifier selection (DCS), classifier structuring and grouping and hierarchical mixture of experts (HME) are considered as from classifier fusion approaches where the fusion

methods work on improving classifiers rather than the results by implementing optimisation techniques on the classifiers.

- ***Dynamic classifier selection***

  Dynamic classifier selection is considered as one of the most straightforward classifier fusion techniques as it applies more than classifier and selects only one classifier, which leads to the best performance. Consequently, the output of DCS is only one classifier rather than a mixture of classifiers as it follows the best local classifier method. One of the widely used technique in DCS is dynamic classifier selection local accuracy (DCS-LA) process where the input is divided into partitions, and the best local classifier for each partition is estimated. Furthermore, the final decision can be computed by allocating an unknown sample to a partition, and the best local classifier for this partition represents the final decision. Another DCS scheme is to estimate a regression model for each partition in the training phase then a dynamic decision function is assigned to the testing phase. DCS is a time-saving and straightforward classifier fusion technique. This approach was used in this study with describing the results in chapters 4, 5, 6 and 7.

- ***Classifier structuring and grouping***

  The fusion methodology implements different functions between classifiers. one of the most common technique is organising combination function in either parallel or sequential order and get their results simultaneously in the case of parallel arrangement only. Besides, another more sensible strategy where all classifiers are arranged into sets and different fusion approaches are applied to each set individually. Additionally, the classifiers can be maintained in a hierarchical order, and different fusion functions can be applied on each stage including dynamic classifier selection one. Accordingly, various algorithms for combining classifiers and fusion techniques can be implemented into different methods. However, the variety of combined classifiers implemented

techniques and the training data should be taken into consideration as these factors may significantly affect the results [278]. Moreover, the fusion algorithm can achieve a substantial improvement in the results through minimising the uncertainty in the information [279].

- ***Hierarchical mixture of experts***

  The hierarchical mixture of experts is another technique for classifier fusion where the classifiers are arranged in a tree structure such that each leave represents an expert. The method of structuring classifiers can affect performance results. The output from each node is divided then gathered by a gating network. This approach recalls Bayes rule by increasing the posterior probability then adjust the combination of classifiers. HME achieved a superior performance when compared to backpropagation neural networks in the field of robotics dynamics [272]. The only drawback of HME is that it may lead to a very complex structure of classifiers combination in the instance of a large dataset.

3.10 Summary

The major objective of this study was introduced in this chapter. The topics of feature selection and deep learning with neural networks were intensely discussed. Various types of neural networks strategy were presented. Signal representation as spectrogram, wavelet and wavelet packet were demonstrated along side data representation techniques like data augmentation and tensor. Classifier fusion was finally discussed, as it will be used in the proposed models detailed in the next chapters. Chapter 4 will present and discuss the implementation of autoencoder as deep learning to classify ten finger movements. Chapter 5 will illustrate the benefits of applying principal component analysis and independent component analysis as feature learning methodologies to classify ten finger gestures. Chapter 6 will discuss the results of employing self-organising map networks for the purpose of deep learning execution to classify ten finger movements. Chapter 7 proposes deep learning as a tool for different biosignal data. It achieved excellent results by virtue of its data augmentation and tensor representation with classifier

fusion. Lastly, Chapter 8 will have the thesis summary, recommendations for future work and conclusion.

CHAPTER 4

Evaluation of Auto Encoder Deep learning for biosignal

In this chapter we develop a deep learning model that is capable of achieving high accuracy results and of learning features without external intervention. As feature extraction plays a role inbiosignal processing, we need to choose one of two paths to identify and select features in any system. The most popular approach is to choose features manually, which heavily depends on user experience and the field of application. The other path is feature learning, which depends on training the system to recognise and pick the best features that match the application. The overall concept of feature learning is to create a model that is expected to be able to learn the best features without any human intervention. Here, autoencoders will be utilised as a feature learning algorithm to train the recommended model to extract useful features from surface electromyography signals. Deep learning method will be presented by using autoencoders to learn features. Wavelet packet, spectrogram, and wavelet approaches will be employed to represent the surface electromyography signal in the recommended model. Then, the newly represented biosignal will be fed to a stacked autoencoder (2 stages) to learn features. Finally, the behaviour of the proposed algorithm will be estimated by using different classifiers such as the extreme learning machine, support vector machine, and softMax layer. The rectified linear unit (ReLU) will be applied as an activation function for the extreme learning machine classifier, besides existing functions such as sigmoid and radial basis functions (RBF). Confidence intervals and analysis of variance will both be estimated for the different classifiers. As a comparative study, the signal representations will be superimposed to compare the results with each representation individually. The classifier fusion layer will be implemented to select the classifier that will lead to the best accuracy' values for both testing and training. The proposed model and the previous work that was done will be explained in Section 4.2. The experimental setup along with the results will be presented in Section 4.3. A discussion will be provided in Section 4.4, followed by a summary in Section 4.5.

4.1 Implementation of Auto Encoder Deep learning for biosignal

In this section, the suggested autoencoder deep learning model will be explained. Figure 4.1 shows the procedures of the recommended system.

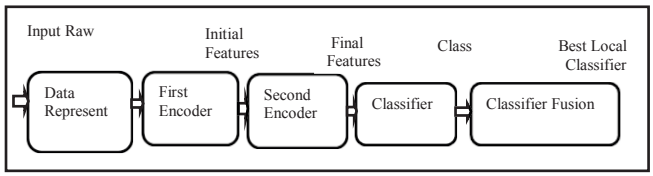


Figure 4. 1: Procedures for sparse autoencoder signal representation

In the proposed model, the autoencoder is a feed-forward neural network that is used in feature learning. The implemented neural network is trained to learn features and produce them as its output, rather than generating classes as in case of using the classification ability of the implemented neural network. A stack autoencoder is implemented, which consists of two successive encoder stages. The input to the encoder is the data whereas the output is the features or representations. The classifier uses features produced from the encoder as an input, its output is the classes' equivalent to the classes of the input data [280]. In the same context of feature learning, the autoencoder generates useful features, rather than producing classes, by decreasing the dimension of the input data. However, the new lower dimensional data will be dealt with as the new features that contain essential and discriminative information on the data, which will help in better classification results. Sparse autoencoder enhanced to leverage the availability of discriminative data as it is trained to learn the useful features. As a comparative study, the three data representations are superimposed and input to the first layer of the autoencoder as shown in Figure 4.2

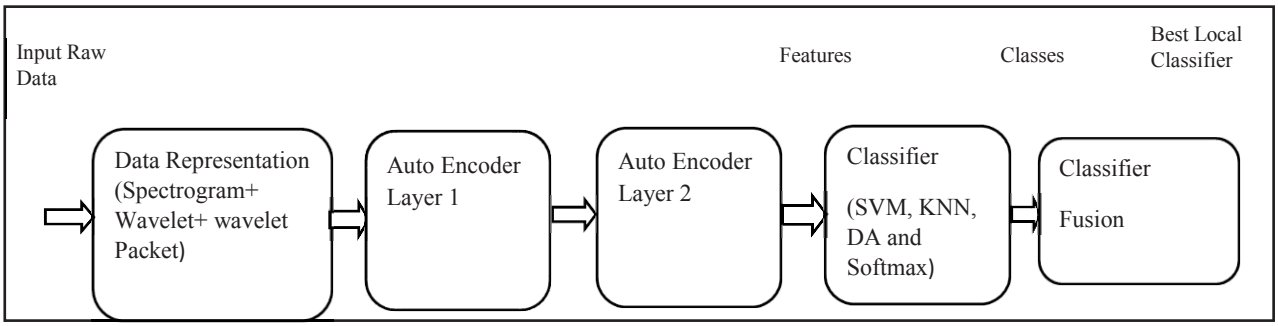


Figure 4. 2: Procedures of superimposed signal representations suggested model

In the suggested model shown in Figure 4.2, the biosignal is represented by using spectrogram, wavelet and wavelet packet methods simultaneously and fed to the cascaded encoder layers. The classifier layer followed the second layer of the autoencoder for evaluating overall system performance. K nearest neighbours (KNN), support vector machine (SVM), softmax and discriminant analysis (DA) are used for the classification task. A classifier fusion layer is then employed to select the best local classifier. In brief, this study aims to present and analyze the planned model by introducing a surface electromyography signal that was previously collected by my Dr Khairul Anam. In this study, trials are performed to establish the feasibility of deep learning in the field of biosignals and to enumerate the techniques that might be employed to improve the system behavior. The effectiveness of both different signal representations along with classifier fusion will be highlighted. Moreover, this effectiveness on enhancing the suggested system performance and promoting the results to a higher accurate level will be investigated.

4.1.1 Previous work

In this research, we will suggest a deep learning system that will be capable of providing essential features from the input signal without recourse to traditional feature extraction and reduction algorithms. The suggested system will be skilful in classifying the ten hand finger motions. The classification of different Finger motions was discussed earlier in many published scientific types of research. The early pattern recognition for finger movements was proposed in [281] where the researchers

suggested using neural networks in analysing and classifying the introduced EMG pattern. They classified both finger movement and joint angle associated with moving fingers. Later, in [282] the authors investigated and optimised configurations between electrode size and arrangement to achieve high classification accuracy. Then, in [283] the researchers gave more attention to selecting extremely discriminative features by employing Fuzzy Neighbourhood Preserving Analysis (FNPA), where the main purpose of this technique is to reduce the distance between the samples that belong to the same class and maximise it between samples of different classes. Other researchers explored traditional machine learning algorithms. And, used time domain features and implement support vector machine, linear discriminant analysis and k-nearest neighbours as different classifiers. Then, they took advantage of genetic algorithms to search for redundancy in the dataset and selected features [284]. In the same context, authors proposed an accurate finger movement classification system by extracting time domain-auto regression features, followed by reducing features by using orthogonal fuzzy neighbourhood discriminant analysis technique. And they implemented linear discriminant analysis as a classifier [285]. Other researchers suggested an accurate pattern recognition system for finger movement by extracting 16 time domain features. The extracted features process the electromyography signals and implementing two-layer feedforward neural networks as classifiers [286]. In contrast, effort and time spent in feature extraction and reduction motivated the introduction of deep learning. Many researchers published valuable achievements in deep learning for biomedical signals. An extensive review study was presented on different types of research that used deep learning in health field [287]. The common factor in each study was the implementation of neural networks to learn features from input biosignals. In the same context, researchers proposed a model by using convolutional neural networks to convert the information which was given by a wearable sensor into highly related discriminative features [288]. Other research presented a deep learning record system that predicted future medical risk automatically after extracting essential features by implementing

convolutional neural networks [289]. Also, researchers implemented a system that was used to extract shallow features from wearable sensor devices, then the features were introduced to convolutional neural networks, and finally to the classifier layer [290]. Based on the above, we can conclude that deep learning is an initial step towards implementing a self-learning system by using neural networks. In our proposed system, neural networks will be implemented in the form of a two stage autoencoder that reads biosignals represented biosignal by either spectrograms, wavelets or wavelet packets. Different classifiers to evaluate the proposed system behaviour will be used. Finally, a classifier fusion layer will be added, which will follow the best local classifier methodology. Adding classifier fusion will be a promising contribution to the accuracy. Moreover, both confidence intervals and analysis of variance will be estimated for different classifiers.

4.2 Experiments and Results

In this part, the data acquisition methods we followed will be expressed more extensively. The experiment setup, including biosignals filtering and amplification will be revealed. The autoencoder layers' settings, classifiers and classifier fusion layer will be presented. And simulation outcomes will be shown and discussed.

4.2.1 Data Acquisition

Biosignals were obtained from nine healthy subjects: seven males and two females. The age of participants was between 24 years and 60 years. No subjects were suffering from any muscle disorder and they were normally limbed. The participants' arms were fixed to avoid motion, which would affect the acquired signal quality. Surface electromyography signals were read by using a FlexComp Infiniti™ device. Two sensors were placed on the forearm of the participant of type T9503M. The placement of two electrodes on a participant's forearm is as shown in Figure 4.3.

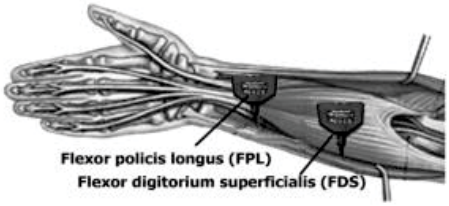


Figure 4. 3: Placement of electrodes [75]

The collected signal was amplified by a gain of 1000 and sampled at frequency 2000 Hz. The obtained signal was processed in Matlab (R2015a) that was installed on a desktop computer with Intel Core i5-6500 and CPU 3.20 GHz. The computer had 8 GB of RAM and running the Windows 7. A bandpass filter, Butterworth third order, was implemented for the frequency range from (20 to 500 Hz). A notch filter was applied of second order at frequency 50 Hz to suppress any 50 Hz interference signals.

The electromyography signal was used to categorise between predefined ten-finger motions, as shown in Figure 4.4, via the suggested model.

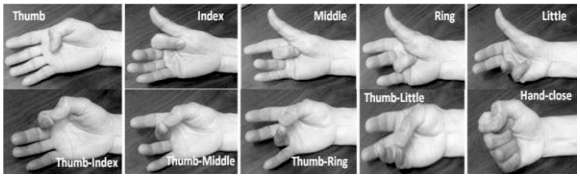


Figure 4. 4: Targeted ten different finger motions [75]

Each participant performed one finger movement for five seconds then had a rest for another five seconds. Each finger motion was repeated six times. The same sequence was repeated for the second finger activity. Three-fold cross-validation was applied to the electromyography signal. Accordingly, 2/3 of the collected data were assigned to the training set whereas the remaining 1/3 was used as the testing set. The average training or testing accuracy was estimated by simulating our proposed model for every subject individually and then summing the accuracies for all subjects and dividing the result by the total number of subjects.

4.2.2 Signal Representation

Three signal representations were applied to the raw biological data to ensure fidelity and precision of the biosignals. Introducing raw data directly in the first autoencoder stage resulted in accuracy less than 50%. The first data representation was the spectrogram for the raw biosignal, whereas the second signal representation used was the wavelet of the signal. As an improving step, we scaled the wavelet signal by five, which in turn enhanced the results as shown in Table 4.1. As a comparative study, we utilised wavelet packets for the signal representation. The signal can be represented in both the time and frequency domain simultaneously. This representation added fidelity to the signal due to its robust representation. In this study, two paths were implemented. The first path was applying each signal representation individually and estimating the accuracies for each case, and the second path was to superimpose the three representations and calculate the accuracy.

4.2.3 Results

For both models, 400 nodes were used for the first layer of autoencoder and 300 nodes for the second one. The number of nodes were selected after applying different experiments to compromise between simulation time and accuracy. The transfer function of the encoder was the pure linear type. Table 4.1 shows autoencoder feature learning testing and training accuracy for the individually implemented signal representations where the biosignal was represented by spectrogram, wavelet, wavelet scale 5 and wavelet packets. Three different classifiers were applied. The first was a SoftMax layer whereas the second classifier was an extreme learning machine. The performance of various activation functions for the extreme learning machine classifier were examined, including sigmoid, the rectified linear unit and radial basis functions. The third classifier was the support vector machine. The performance of linear support vector machine, quadratic support vector machine, cubic support vector machine, fine Gauss support vector machine, medium Gauss support vector machine and coarse

Gauss support vector machine were studied. The best support vector machine that showed the greatest classification ability was selected to be the implemented support vector machine.

Table 4. 1: Autoencoder testing and training accuracy for individual signal representation

Signal Representing	Average Training Accuracy	Average Testing Accuracy	Classification Algorithm	Simulation Time
Spectrogram	95%	73%	SoftMax Layer	830.93 Seconds
Spectrogram	95.5%	79.14%	ELM (Sigmoid)	379.10 Seconds
Spectrogram	95.5%	82.78%	ELM (ReLU)	379.54 Seconds
Spectrogram	97.237%	83.56%	ELM (RBF)	353.756 Seconds
Spectrogram	89.48%	77.27%	Cubic SVM	1000.409 Seconds
Wavelet	91.42%	45.71%	SoftMax Layer	210.04 Seconds
Wavelet	79.95%	59.88%	ELM (Sigmoid)	265.23 Seconds
Wavelet	83.62%	64.73%	ELM (ReLU)	276.149 Seconds
Wavelet	81.98%	62.70%	ELM (RBF)	257.83 Seconds
Wavelet	70.29%	52.29%	Linear SVM	341.57 Seconds
Wavelet (Scale 5)	98.85%	82.13%	SoftMax Layer	668.355 Seconds
Wavelet (Scale 5)	95.65%	85.416%	ELM (Sigmoid)	402.647 Seconds
Wavelet (Scale 5)	96.98%	86.827%	ELM (ReLU)	276.697 Seconds
Wavelet (Scale 5)	95.59%	85.58%	ELM (RBF)	444.067 Seconds
Wavelet (Scale 5)	96.55%	83.85%	Cubic SVM	495.124 Seconds
Wavelet Packet	98.69%	84.176%	SoftMax Layer	429.52 Seconds
Wavelet Packet	93.3%	86.79%	ELM (Sigmoid)	419.27 Seconds
Wavelet Packet	96.42%	89.41%	ELM (ReLU)	419.27 Seconds
Wavelet Packet	95.59%	87.78%	ELM (RBF)	540.22 Seconds
Wavelet Packet	98.16%	89.707%	Quad SVM	579.713 Seconds

From the results shown above, it can be found that the classification ability of the extreme learning machine was outstanding in this application for all signal representations except for wavelet packets. Both quadratic support vector machine, and extreme learning machine with the rectified linear unit as an activation function, showed very close performance to wavelet packets. Extreme learning machine was improved when the sigmoid activation function was replaced by radial basis functions

and the rectified linear unit. The rectified linear unit activation function for extreme learning machine achieved superior behaviour than radial basis functions and sigmoid activation functions for wavelet, wavelet scale 5 and wavelet packet. However, the rectified linear unit offered better performance than sigmoid and lower accuracy than radial basis function for spectrogram.

Cubic and Quad support vector machine results in good testing accuracy for wavelet scale 5 and wavelet packet only. Simulation time for the support vector machine is relatively longer than other compared classification algorithms. SoftMax layer resulted in very poor classification for the wavelet signal representation, as the testing accuracy was less than 50%. Softmax proved its classification ability for wavelet scale 5 and wavelet packet.

Figure 4.5 illustrates different P values for different classifiers resulted from ANOVA calculation, and Figure 4.6 shows confidence intervals for each classifier where it was calculated twice. One was with wavelet and the other was without wavelet. The yellow bars were calculated for different classifiers with three signal representation methods (wavelet scale5, wavelet packet and spectrogram). Blue bars were estimated for different classifiers with four signal representation methods (wavelet, wavelet scale5, wavelet packet and spectrogram). The narrowest interval was 2.53% for the extreme learning machine. The widest was 6.10% for support vector machine, and softmax layer interval reached 5.77%.

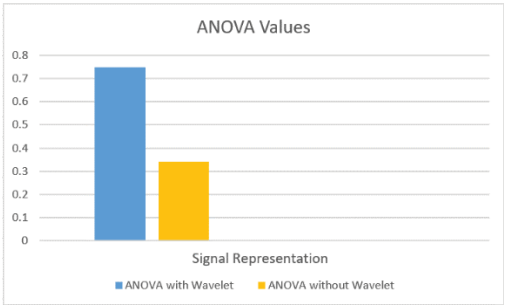


Figure 4. 5 : ANOVA values for different Classifiers for individual signal representation

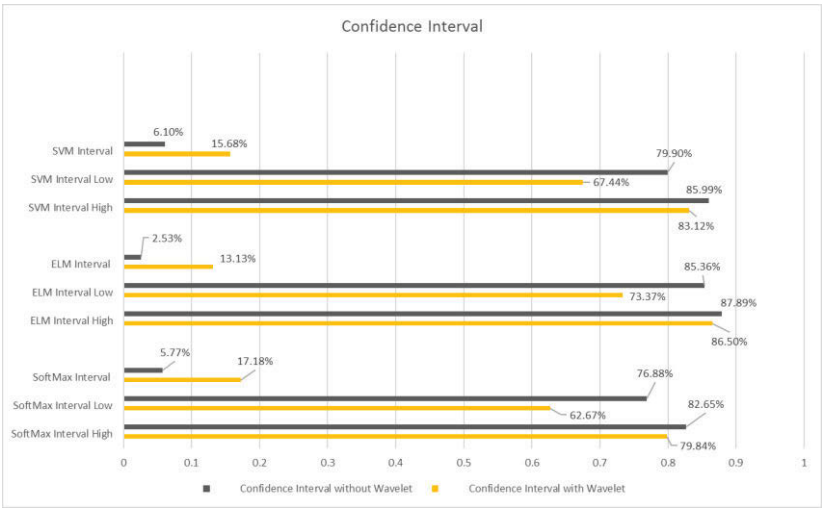


Figure 4. 6: Confidence Intervals for used classifiers for individual signal representation

A layer of classifier fusion was concatenated after classification layer. The function of this added layer is to nominate the best-implemented classifier based on the outcomes of accuracies values. This added classifier fusion layer in return enriched the accuracies as displayed in Table 4.2. On the other hand, adding the classifier fusion layer increased the simulation time.

Table 4. 2: Autoencoder classifier fusion testing and training accuracy for individual signal representation

Signal Representing	Average Training Accuracy	Average Testing Accuracy	Classification Algorithm	Simulation Time
Spectrogram	99.53%	91.05%	Classifier Fusion	2118.846 Seconds
Wavelet	96.99%	86.80%	Classifier Fusion	745.67 Seconds
Wavelet (Scale 5)	99.24%	89.02%	Classifier Fusion	1378.41 Seconds
Wavelet Packet	98.70%	92.25%	Classifier Fusion	1656.724 Seconds

The same signal was implemented for both models (individual signal representation and superimposed signal representation). A sample of this biosignal is displayed in Figure 4.7. Figure 4.7 represents the electromyography signal collected for ten-finger gestures, where the y-axis represents the EMG signal in volts versus time in milliseconds.

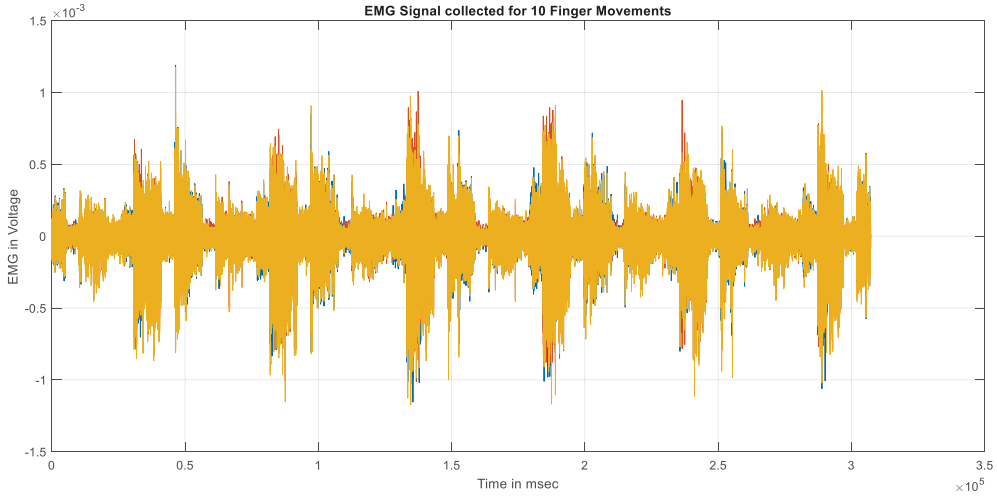


Figure 4. 7: Electromyography signal for ten-finger movements

Table 4.3 shows the results of implementing two cascaded autoencoder layers as the deep learning technique for three-fold cross-validations. Table 4.4 exhibits the results of executing two cascaded autoencoder layers as the deep learning technique for ten-fold cross-validations. The biosignal was represented in spectrogram, wavelet and wavelet packet representations simultaneously.

Table 4. 3: Training and testing accuracies for three-fold cross validations for superimposed signal representation

Training Accuracy	Testing Accuracy	Classifier
98.01%	86.55%	Support Vector Machine
97.58%	84.28%	Discriminant Analysis
98.78%	87.63%	K nearest Neighbours
97.35%	81.25%	SoftMax

Table 4. 4: Training and testing accuracies for ten-fold cross validations for superimposed signal representation

Training Accuracy	Testing Accuracy	Classifier
97.89%	86.21%	Support Vector Machine
97.32%	83.78%	Discriminant Analysis
98.01%	87.24%	K nearest Neighbours
96.86%	80.67%	SoftMax

For the three-fold cross-validations, the k nearest neighbours showed the highest testing accuracy while softmax showed the lowest capability of classifying the input biosignal. Support vector machines showed a high testing accuracy whereas discriminant analysis

showed a moderate one. The behaviour of the proposed model did not change significantly towards the ten-fold cross-validations as shown in Table 4.4. The raw data were introduced directly into the first layer of the autoencoder, without applying any signal representation. The testing accuracy was less than 50%. This low value ignited the idea of applying different types of signal representations.

Classifier fusion implementation increased the testing accuracy to reach 90.75% for three-fold cross-validations and 90.50 % for ten-fold cross-validations where the classifier fusion layer followed the best local classifier technique.

The simulation time that was consumed to train the network was on the average 450 seconds, however, it took 1.2 seconds to test the trained network.

The testing accuracies for different classifiers were calculated versus window size. The window size values vary from 50 milliseconds to 500 milliseconds. As shown in Figure 4.8, the highest testing accuracy was achieved at window size 200 milliseconds. Furthermore, the four classifiers behave in the same manner with regard to testing accuracy versus window size. Therefore, all of our testing accuracies were calculated at window size 200 milliseconds to obtain the highest testing accuracy.

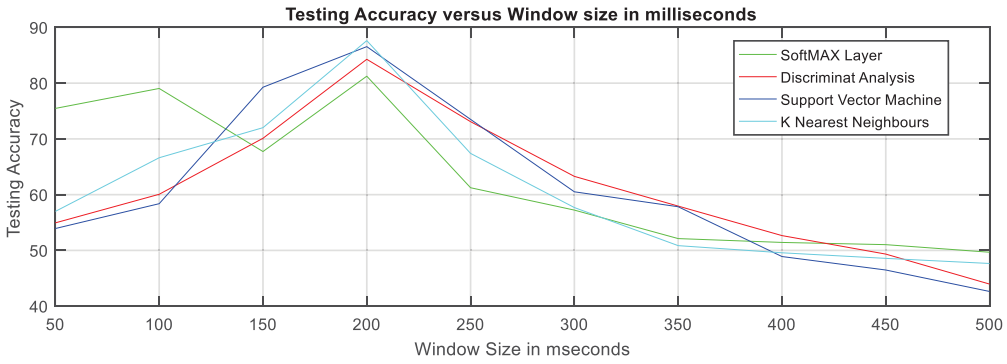


Figure 4. 8: Testing accuracy versus window size for superimposed signal representation

4.2.4 Comparison between the proposed model with other researchers' achievements

In this section, we will compare between the proposed model with other researchers achievements for the same dataset. The same dataset had been used before by researchers in [79] who targeted the same ten finger movements combinations. The

scholars developed an adaptive wavelet extreme learning machine (AW-ELM) for classification. The authors employed a pattern recognition model and compared the results of AW-ELM to both sigmoid and wavelet activation functions for an extreme learning machine. The dataset that was implemented in the autoencoder deep learning model is identical to that utilised in this pattern recognition model where the researchers followed the stages that are shown in Figure 4.9.

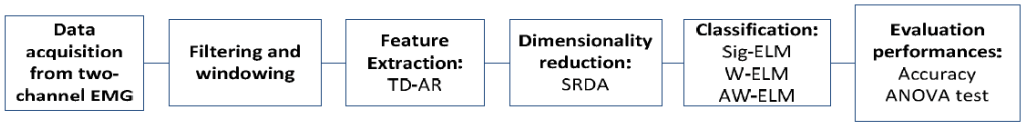


Figure 4. 9: Architecture of pattern recognition with AW-ELM

The two-channel dataset was filtered and windowed with a window length 200 ms and window increment 25 ms. Afterwards, the time domain features were extracted. The authors applied sample skewness (SS), slope sign changes (SSC), number of zero crossings (ZC) and waveform length (WL). Autoregressive (AR) and Hjorth time domain (HTD) parameters were also estimated. The extracted features were reduced by applying spectral regression discriminant analysis (SRDA). After that, the researchers used the reduced features for classification. The authors applied extreme learning machine for classification along with executing adaptive wavelet, sigmoid and wavelets as activation functions. Finally, for evaluation performance and statistical analysis, the P value between AW-ELM and two other classifiers were calculated after estimating ANOVA. The authors changed the number of hidden nodes in extreme learning machine neural networks from 50 to 500. Table 4.5 elucidates the average testing accuracy values for three activation functions while changing the number of nodes in the extreme learning machine.

Table 4. 5: Average testing accuracy for three activation functions

# Hidden Node	Accuracy %		
	W-ELM	AW-ELM	Sig-ELM
50	91.07 ± 0.17	91.57 ± 0.08	91.65 ± 0.08
75	91.56 ± 0.14	91.93 ± 0.14	91.97 ± 0.10
100	91.79 ± 0.10	92.05 ± 0.09	92.01 ± 0.08
125	91.90 ± 0.08	92.08 ± 0.09	92.03 ± 0.10
150	91.49 ± 0.11	92.06 ± 0.10	92.04 ± 0.10
175	91.98 ± 0.09	92.06 ± 0.09	92.04 ± 0.08
200	91.99 ± 0.08	92.04 ± 0.08	92.01 ± 0.06
500	91.79 ± 0.08	91.56 ± 0.06	91.37 ± 0.06

Table 4.6 shows the processing time in seconds for training and testing sets. Table 4.7 illustrates the P value for AW-ELM with other activation functions.

Table 4. 6: Average processing time for ELM with different activation functions

# Hidden	Training Time (s)			Testing Time (s)		
Node	W-ELM	AW-ELM	Sig-ELM	W-ELM	AW-ELM	Sig-ELM
50	0.16 ± 0.01	0.19 ± 0.02	0.12 ± 0.00	0.03 ± 0.00	0.06 ± 0.00	0.01 ± 0.00
75	0.28 ± 0.01	0.33 ± 0.01	0.19 ± 0.00	0.06 ± 0.01	0.07 ± 0.00	0.02 ± 0.00
100	0.38 ± 0.02	0.45 ± 0.02	0.27 ± 0.01	0.07 ± 0.00	0.10 ± 0.00	0.02 ± 0.00
125	0.59 ± 0.05	0.70 ± 0.06	0.48 ± 0.07	0.09 ± 0.00	0.14 ± 0.00	0.03 ± 0.00
150	0.71 ± 0.01	0.81 ± 0.01	0.51 ± 0.01	0.12 ± 0.00	0.19 ± 0.01	0.04 ± 0.00
175	0.93 ± 0.06	1.06 ± 0.05	0.69 ± 0.05	0.14 ± 0.00	0.22 ± 0.00	0.04 ± 0.00
200	1.07 ± 0.08	1.22 ± 0.05	0.85 ± 0.06	0.17 ± 0.00	0.26 ± 0.00	0.05 ± 0.00
500	4.20 ± 0.08	5.42 ± 0.08	2.82 ± 0.10	0.75 ± 0.01	1.28 ± 0.01	0.12 ± 0.00

Table 4. 7: P-value for AW-ELM and other activation functions

#Hidden Node	***P*-value**	
	AW-ELM & W-ELM	**AW-ELM & Sig-ELM**
50	0.0000	0.0000
75	0.0000	0.1283
100	0.0000	0.0006
125	0.0000	0.0610
150	0.0000	0.3477
175	0.0021	0.5746
200	0.0098	0.0552
500	0.0000	0.0000

The highest testing accuracy that was achieved by the AW-ELM pattern recognition model was 92.08 ± 0.09, whereas the deep learning autoencoder attained accuracy 92.25% for the testing set and 98.70% for training set. In AW-ELM, the feature

extraction was manual and differed according to the application. For the deep learning model, the suggested scheme was able to learn features by itself regarding the application, which in turn saved user's time and effort. Regarding the simulation time, AW-ELM authors achieved 0.70 ± 0.06 seconds for training and 0.14 ± 0.00 seconds for testing. The deep learning autoencoder consumed a long simulation time in the training phase, 1656.724 seconds for training the network, and spent less than 1.5 seconds for testing. The P-value for the deep learning autoencoder implemented classifiers showed values more than 0.05, which meant that there was no substantial difference between the used classifiers. For the AW-ELM pattern recognition scheme, P values attained were 0.0610 for AW-ELM and sig-ELM. This value meant that there was no considerable difference between the two employed activation functions. Besides, P-value reached 0.0000 for AW-ELM and W-ELM that showed a significant difference between two activation functions. As AW-ELM achieved a substantial improvement in the accuracy values.

4.3 Discussion

Sparse autoencoder is an algorithm with only one hidden layer. Therefore, to establish the concept of deep learning, and take advantage of stacking more than a layer, a stacked autoencoder was implemented that led to verifying the deep learning concept and enriching the results accuracy. In addition, applying some signal representations such as spectrogram, wavelet and wavelet packets, instead of using raw biosignal. And then the output of these signal representation was introduced to the first stage of autoencoder enhanced the performance of the system. The extreme learning machine showed satisfactory performance on the level of testing accuracy and simulation time. Softmax layer classification resulted in the most mediocre testing accuracy although it consumed longer simulation time than the extreme learning machine. Support vector machines produced excellent testing accuracy but consumed more simulation time. Applying signal representations such as spectrogram, wavelet or wavelet packets

improved both training and testing accuracy. As both accuracies were much less than 50% when we fed first stage autoencoder by raw data. Multiplying wavelet scale by 5 enhanced the results. In conclusion, applying any signal representation, either in the time domain, frequency domain, or both, had a good impact on our training and testing accuracies. The rectified linear unit was introduced as an activation function for extreme learning machines besides already existing functions such as radial basis function and sigmoid. The rectified linear unit was superior in its testing accuracy to both radial basis function and sigmoid one for wavelet, wavelet scale 5 and wavelet packet signal representations. It resulted in lower testing accuracy than radial basis functions but better than the sigmoid for spectrogram signal representation.

Calculating ANOVA gave us an indication on how close or far was different classifiers. The computation of confidence interval with confidence score 60% gave us the upper and lower accepted accuracies. Adding a classifier fusion layer was very helpful in improving the percentages of our accuracy in either the training set or the testing set. However, it consumed significant simulation time in comparison to that consumed without fusion layer.

For the superimposed signal representation model, the raw biosignal was represented by spectrogram, wavelet and wavelet packet simultaneously and then the represented signal was introduced to the first layer of autoencoder. The output of this layer would be treated as features to be considered as the inputs of the second layer of autoencoder. Afterwards, the suggested model was evaluated by four classifiers (KNN, Softmax, SVM and DA). K nearest neighbours showed superior performance to the other three implemented classifiers where the testing accuracy reached 87.63%. Support vector machine was a good classifier as well in which the testing accuracy was 86.55%. Softmax resulted in the least classification capability as it was 81.25%. Classifier fusion improved the result as the testing accuracy achieved 90%. The 200 ms window size led to obtaining the highest testing accuracy. In conclusion, deep learning was an initial step towards saving effort and time wasted in extracting and reducing features as it learnt, by itself, the best features suitable for the application under examination. In addition, since feature extraction and reduction methods varied according to the application, feature extraction and reduction algorithms were not fixed and needed more experience. In other words,

deep learning systems should be adaptable to any set of data if the data were accurate and well represented. This objective brought a new challenge on the scene in regards to representing the data. The data should be represented in a high precision way to expect a good result from implementing deep learning techniques.

4.4 Summary

The implementation of two-stage autoencoder, for the purposes of deep learning, was presented. The raw biosignal was introduced directly to the first stage of autoencoder and led to an accuracy less than 50%. However, by executing different signal representations either individually or simultaneously boosted the accuracy to values more than 80%. Many classifiers were recalled support vector machine, and k nearest neighbour classifiers were superior for individual signal representation deep learning and simultaneous signal representation deep learning respectively. The best local classifier technique, based on dynamic classifier selection criteria, was followed to select the highest accuracy achieved by a classifier. The implementation of classifier fusion layer boosted the results, for some signal representations to be over 90% and for the simultaneous signal representation deep learning model as well. Both the ANOVA and confidence interval was estimated for individual signal representation deep learning model. As it was mentioned earlier that the primary objective of this research is to implement a robust deep learning model that should be capable of learning features for any biosignal input and achieve high accuracy values, therefore, all the following chapters will be in the light of this context. The executed deep learning autoencoder model showed a superior behaviour regarding the accuracy value. Furthermore, the proposed deep learning scheme did not require any manual feature extraction algorithms, as the suggested system was able to learn features by itself. However, the planned deep learning model consumed long simulation time in training the network while it spent a short time in the testing phase. The long simulation time was the principal motivation that led us to implement principal and independent component analyses for feature learning as will be illustrated in the following chapter. Chapter 5 will

discuss the implementation of principal and independent component analysis for feature learning purposes. Moreover, chapter 6 will demonstrate the evaluation of self-organising map network as a deep learning suggested model. In addition, chapter 7 will illustrate the execution of data augmentation along with tensor representation for deep learning implementation. Eventually, chapter 8 will summarise the thesis and will recommend future work

CHAPTER 5

Evaluation of PCA and ICA Feature Learning for Biosignals

Supervised learning is considered to be a highly limited technique despite being applied in different applications. Most of the applications require manual feature extraction by using different algorithms. Whenever the feature representation is good, classification error should be expected to be low. A good representation means that the primary objective is to represent the data by using discriminative features. However, manual feature extraction is still exhausting and time-consuming. Many suggested feature learning techniques may be applied to enhance feature representation automatically and save time and effort. The principal evaluation of the performance of applied feature learning techniques is the classification error. In this chapter, principal component analysis and independent component analysis will that implements as feature learning techniques to teach the model learn the features from the input data. A deep learning approach will be proposed by implementing either PCA or ICA to learn features. In the proposed model, the raw data will be read then represented by using different signal representations such as the spectrogram, wavelet and wavelet packets.

The new represented data will be introduced to the principal component analysis layer or the independent component analysis layer to generate features. The performance of the suggested scheme will be evaluated by applying different classifiers such as the support vector machine, extreme learning machine, and discriminant analysis classifiers. To improve the results, a classifier fusion layer will be implemented to select the most accurate result for both the training and the testing set. Moreover, the Manhattan index will be estimated for the PCA learnt features to reduce the number of features used. Feature reduction is done by selecting the highest 50 Manhattan index features. The objective of assessing Manhattan index is to use a lower number of features, which in turn means lower simulation time while maintaining the same accuracy values. A discussion of the proposed model will be presented in Section 5.2. The experimental

setup along with the results will be explained in Section 5.3. Conclusions will be presented in Section 5.4, followed by a summary in Section 5.5.

5.1 Implementation of PCA and ICA feature learning for biosignal

In this section, the suggested feature learning model will be explained. Figure 5.1 shows the procedures of the recommended system.

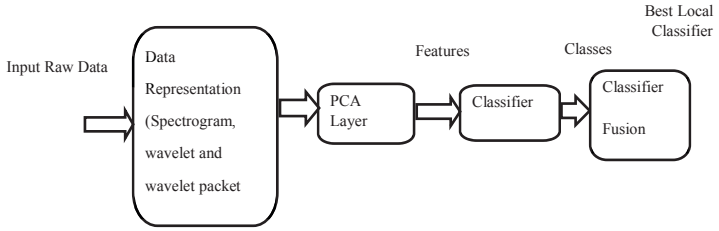


Figure 5. 1: Procedures of the suggested PCA feature learning model

In the proposed model, PCA is a dimensional reduction layer that is used in feature learning. The raw data are introduced to the data representation phase where the output is the represented data. Three data representation schemes are implemented. The first is the spectrogram, the second is the wavelet, and the third is the wavelet packets. The input to the PCA layer is the represented data version of the raw data whereas the output is the features. The classifier uses features produced from the PCA layer as an input whereas its output is the classes' equivalent to the input data. A classifier fusion layer is employed to select the best local classifier. The PCA layer is replaced by an ICA layer as shown in Figure 5.2.

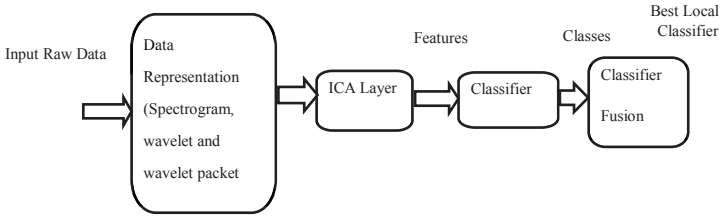


Figure 5. 2: Procedures of the proposed ICA feature learning model

5.1.1 Independent component analysis

ICA was mentioned in brief in Chapter 3. However, more detailed information on ICA will be explained in this section. ICA is a mathematical or computational transformation that seeks a new linear transform that decreases linear dependency between two components [291]. ICA is convenient in separating linearly combined independent sources. For example, the electroencephalograms (EEG) signals that can be collected using multi-electrode from the scalp. One of the favourite applications of ICA is the "cocktail party effect". The cocktail party effect is merely separating the sounds of people who are talking simultaneously in a noisy room [292]. Listening to the voice of an individual in a noisy place can be done merely by assuming that each subcomponent is a non-Gaussian distributed signal and the subcomponents are linearly independent. The cocktail party effect gives the opportunity to focus on one speaker only within a noisy room with mixing sounds in the background [293]. Two main problems are controlling the cocktail party effect. The first is recognising a particular voice within mixing voices, in other words, which signals are being used by humans to distinguish a specific voice within a noisy room. This problem is solved by making a machine that mimics the human abilities. The second problem is the composition of the previously selected signals, where the listener will be able to differentiate between the mixed voices [294].

5.1.2 Manhattan index

The Manhattan index is a suggested technique that can be used in selecting features to avoid redundancy and save simulation time.

Manhattan Index= Manhattan distance/Standard deviation

The Manhattan distance is just the distance between the projection of each point on the horizontal and the vertical axis.

The Manhattan index should be calculated for each feature. Then, the features will be selected according to the value of the index. The top 50 features will be selected. Those

features correspond to the highest Manhattan index value. By combining these 50 features with each other, we guarantee that redundancy of features is in its minimum value.

5.2 Experiments and Results

In this section, the PCA and ICA layers' settings, classifiers and classifier fusion layer will be presented. The simulation outcomes will be presented and discussed. The data acquisition, filtering, amplification, and experimental setup were identical to that explained in Chapter 4.

5.2.1 Results

Three classification algorithms were implemented, where the first was the support vector machine, the second was the extreme learning machine and the third was the discriminant analysis. The accuracy for six different types of SVM (Linear SVM, Quad SVM, Cubic SVM, Fine Gauss SVM, Medium Gauss SVM and Coarse Gauss SVM) was compared. Also, the results of linear and quadrature discriminant analysis were compared. Table 4.1 shows the accuracy values for both training and testing sets for the PCA layer feature learning for the three classifiers.

Table 5. 1: PCA training and testing accuracy

Signal Representation	Average Training Accuracy	Average Testing Accuracy	Classification Algorithm
Spectrogram	99.5324%	80.2595%	Support Vector Machine
Spectrogram	90.984%	77.527%	Extreme Learning Machine
Spectrogram	90.437%	77.368%	Discriminant Analysis
Wavelet	99.7138%	81.124%	Support Vector Machine
Wavelet	94.0476%	86.607%	Extreme Learning Machine
Wavelet	88.666%	81.128%	Discriminant Analysis
Wavelet Packet	98.4172%	88.9014%	Support Vector Machine
Wavelet Packet	98.0166%	87.78%	Extreme Learning Machine
Wavelet Packet	98.42%	88.15%	Discriminant Analysis

It is clear from the results that are shown in Table 5.1 that using the wavelet packets as a signal representation generated a higher testing accuracy than both the wavelet and spectrogram. The wavelet had a higher testing accuracy than the spectrogram and had a lower testing accuracy than the wavelet packets. The support vector machine achieved a higher classification ability for both the spectrogram and the wavelet packets than its classification ability in the wavelet signal representation. On the other hand, the extreme learning machine was superior to the other classification techniques for the wavelet signal representation. The classifier fusion layer was added, where it selected the best local classifier by using dynamic classifier selection algorithm. Both the training and testing accuracies were improved as shown in Table 5.2.

Table 5. 2: Classifier fusion training and testing accuracy for PCA

Signal Representation	Average Training Accuracy	Average Testing Accuracy	Classification Algorithm
Spectrogram	98.014%	82%	Classifier Fusion
Wavelet	98.42%	87.5%	Classifier Fusion
Wavelet Packet	98.5%	90.318%	Classifier Fusion

The testing accuracies were improved by adding classifier fusion, where the wavelet packets testing accuracy exceeded 90%. The wavelet testing accuracy became higher than 87%, and ultimately, the spectrogram became 82%. In comparing Table 5.2 results with Table 5.1 results a conclusion can be reached that the classifier fusion improved results. The only drawback for the classifier fusion was increasing the simulation time as it became relatively higher than that without implementing a classifier fusion. However, the simulation time consumed by PCA as feature learning is short. The number of features learnt for each data representation is different as shown in Figure 5.3. Where the spectrogram had 387 features, the wavelet had 300 and the wavelet packets had only 60 features. The idea of estimating the Manhattan index for features and selecting the highest 50 index features estimated when the results in Table 5.1 were analysed. Despite having the most abundant number of features for the spectrogram

data representation, the accuracy values were lowest among the other two signal representations. This result established the fact that the larger number of features would not necessarily lead to a higher accuracy value. Moreover, the combination of features should be done while avoiding redundancy.

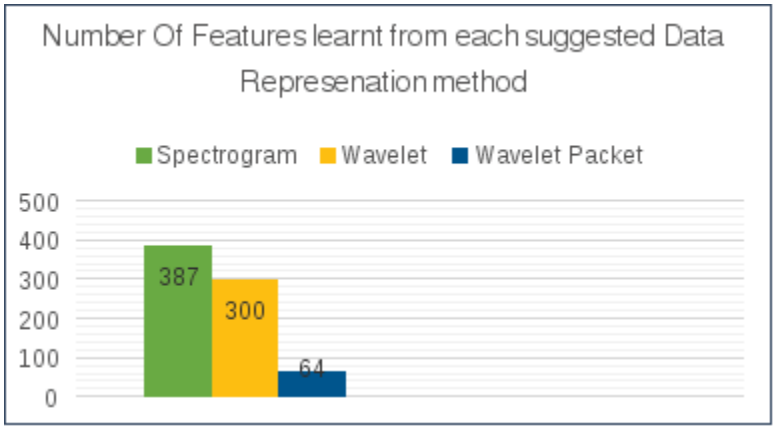


Figure 5. 3: Number of learnt features for each data representation

Finally, the top 50 Manhattan index 50 features were selected from the PCA feature learning algorithm after calculating the Manhattan index for each feature. The SVM was employed as a classifier for 50 features where the results are as shown in Table 5.3.

Table 5. 3: Feature indexing training and testing accuracy

	All Features		50 Features	
Signal Representation	***Average Training Accuracy***	***Average Testing Accuracy***	***Average Training Accuracy***	***Average Testing Accuracy***
Spectrogram	99.5324%	80.2595%	95.076%	78.566%
Wavelet	99.7138%	81.124%	97.493%	80.404%
Wavelet Packet	98.4172%	88.9014%	98.2063%	84.18%

Limiting the number of learnt features did not affect both the wavelet and the spectrogram testing accuracies substantially. However, it was more influential on the wavelet packets testing accuracy as it was reduced by almost 5%.

On the other hand, the PCA layer was replaced by ICA for feature learning where the results are as shown in table 5.4.

Table 5. 4: ICA training and testing accuracy

Signal Representation	Average Training Accuracy	Average Testing Accuracy	Classification Algorithm
Spectrogram	98.5833%	81.8564%	Support Vector Machine
Spectrogram	96.7432%	80.2148%	Extreme Learning Machine
Spectrogram	98.45205%	79.4678%	Discriminant Analysis
Wavelet	98.0557%	84.375%	Support Vector Machine
Wavelet	96.89135%	83.4235%	Extreme Learning Machine
Wavelet	98.0394%	82.52346%	Discriminant Analysis
Wavelet Packet	96.55%	85.249%	Support Vector Machine
Wavelet Packet	98.0166%	85.827%	Extreme Learning Machine
Wavelet Packet	98.54%	84.473%	Discriminant Analysis

The wavelet packets signal representation resulted in the highest testing accuracy for the three classifiers whereas the spectrogram had the lowest testing accuracy.

The classifier fusion layer was added, where it selected the best local classifier by using the dynamic classifier selection algorithm. Both the training and testing accuracies were improved as shown in table 5.5.

Table 5. 5: Classifier fusion training and testing accuracy for ICA

Signal Representation	Average Training Accuracy	Average Testing Accuracy	Classification Algorithm
Spectrogram	98.6%	83.4037%	Classifier Fusion
Wavelet	99.4963%	83.55468%	Classifier Fusion
Wavelet Packet	98.6066%	86.06283%	Classifier Fusion

The classifier fusion layer improved the accuracy values for the three signal representations. However, the wavelet signal representation accuracy was not amended as other two representations whereas the spectrogram along with the classifier fusion, approached the wavelet accuracies. The wavelet packets performance was improved after using the classifier fusion layer. Moreover, wavelet packet was the superior signal representation that matched the application either with implementing classifier fusion layer or not.

5.2.2 Comparison between the proposed model with other researchers' achievements

In this section, we will compare between the proposed model with other researchers achievements for the same dataset. The dataset that was used for the PCA and the ICA feature learning was utilised by the authors in [81]. The researchers implemented a pattern recognition model that was able to expect any shift that occurred to the electrode location as well as a change in muscle strength. Moreover, the authors proposed a scheme that should be capable of adaption any alteration. The authors attempted to reduce the gap between the laboratory results and the real-life application results. The researchers used the online sequential extreme learning (OS-ELM) for classification where the input training data were introduced segment by segment and classification was completed in three phases. The first phase was offline classification, whereas the second one was adaptation and the third phase was online classification. The authors applied the same filtering and sampling rate as the proposed deep learning models used in chapter 4 and 5. Moreover, the researchers applied three-fold cross-validations. The same ten classes that were classified in chapter 4 and 5 were the targeted classes for the researchers of this pattern recognition scheme. The window size was 100 ms, and the authors followed the same scenario regards the extracted and reduced time domain features that were executed in chapter 4. The authors applied sample skewness (SS), slope sign changes (SSC), number of zero crossings (ZC) and waveform length (WL). Autoregressive (AR) and Hjorth time domain (HTD) parameters were also estimated. The extracted features were reduced by using the spectral regression discriminant analysis. The reduced features were used in the offline classification, the first classification phase, where the features were divided into 0.5 seconds for initialisation stage while the rest were employed for sequential phase with an increment 100 ms. Whereas the second phase of classification was the adaptation that was activated with a new session setup. The new session of recording data included acquiring data for a

shorter period, than the first classification stage, to be used in training the OS-ELM to adapt to the new acquaintance in the new session. Moreover, the third classification phase was the online classification were the classification was executed while acquiring the datasets. The authors investigated the average error versus the number of nodes for extreme learning machine neural network for offline classification, which is shown in Figure 5.4.

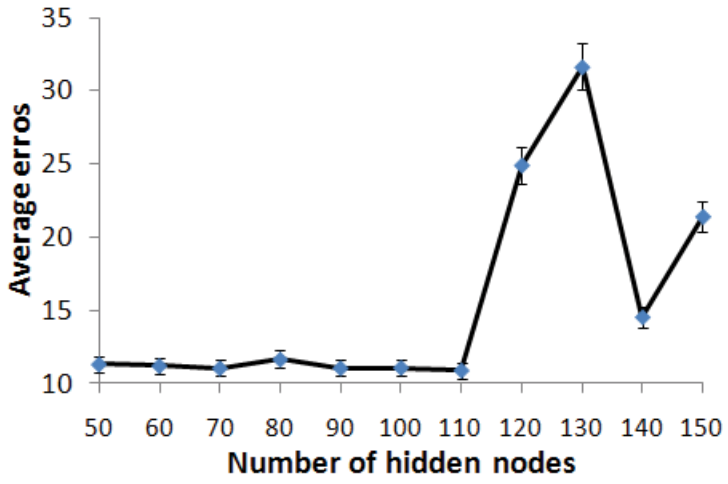


Figure 5. 4: Error rate versus number of hidden nodes for offline classification [81]

The researchers applied daily classification on one subject only for three successive days by implementing the OS-ELM and the radial basis function ELM (RBF-ELM) as shown in Figure 5.5.

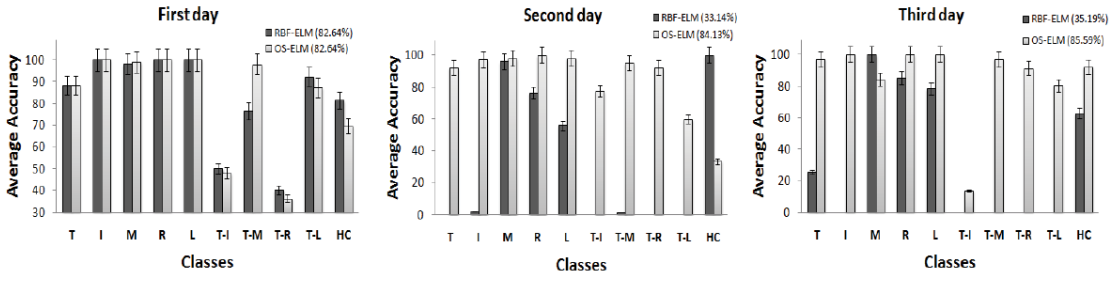


Figure 5. 5: Average accuracy versus ten classes for RBF-ELM & OS-ELM [81]

The average accuracy on the first day was 82.64 %for both the OS-ELM and the RBF-ELM whereas the accuracy was 84.13% for the OS-ELM and 33.14% for the RBF-ELM. Moreover, the accuracy reached 85.59% for applying the OS-ELM and 35.19 % for applying the RBF-ELM. In conclusion, the suggested model showed a stable accuracy range of 82 % - 85% and achieved accuracy value of 35 % for the non-adaptive classifiers as the RBF-ELM. On the other hand, the proposed feature learning model attained a testing accuracy over 90% without applying any feature extraction and feature reduction techniques. Furthermore, the simulation time spent by the feature learning system was concise for both the training and the testing phases. Besides, the recruitment of feature learning guaranteed the learning of best features that matched the application whereas feature extraction algorithm depended on the user's experience and a trial and error procedure.

5.3 Discussion

PCA is a time-saving and easy feature learning technique, where it reduces dimensions of data without too much loss in of information and observations that should be associated with the data.

The spectrogram, wavelet and wavelet packets helped in better representing the raw data. The SVM and ELM achieved good classification ability in the spectrogram, wavelet packet and wavelet respectively.

The classifier fusion layer improved the testing accuracies for the three signal representations used. The classifier fusion algorithm selected the best local classifier.

Limiting the number of learnt features to 50 features did not affect the testing accuracies for both the wavelet and the spectrogram. The testing accuracy difference between combining all learnt features and combining the top 50 Manhattan index features did not exceed 2.5%. In the case of the wavelet packets the testing accuracy difference was almost 5%. On the other hand, the ICA is an uncomplicated feature learning technique which transforms data to a new linearly

independent transform. Adding a classifier fusion layer improved the accuracy regardless of the feature learning algorithm that was applied. As a recommendation for future work, combining all features for the three suggested data representation and selecting the top 50 Manhattan index features, then applying a classifier fusion to attain reaching real simulation time in real hands applications that varies between 150ms to 300 ms besides achieving high testing accuracies rate.

5.4 Summary

In this chapter, PCA and ICA were used for the purpose of feature learning. Three signal representations, the spectrogram, wavelet and wavelet packets, were implemented to enhance the performance of the proposed model. Introducing the raw biosignals directly to either the PCA layer or the ICA layer led to a very low accuracy, less than 50%. The ELM, SVM and discriminant analysis were employed for the classification task. The spectrogram signal representation resulted in the lowest accuracies whereas the wavelet packets led to the highest accuracies. The implementation of the classifier fusion layer was beneficial, from the accuracy point of view, for both the PCA and the ICA proposed models as the accuracies were improved to a level higher than 90% for the wavelet packets in the PCA feature learning model. However, the added classifier fusion layer increased the simulation time because it compared between three classifiers and picked the highest accuracy for each class. Nevertheless, neither the PCA model nor the ICA consumed a long time in the simulation phase so adding a fusion layer did not extend the simulation time. Despite representing the biosignals with a more significant number of features in the spectrogram signal representation, the accuracy was the lowest among other two signal representations, the wavelet and the wavelet packets. This result initiated the motivation to remove the redundant features to save simulation time. The Manhattan index was the procedure for this task where the index was estimated for each feature, and only the highest 50 were included in the simulation. The implemented PCA and ICA feature learning models achieved accepted accuracy values as well as short simulation time for both training and testing

stages. However, improving the performance motivated us to employ self-organising map to achieve higher testing accuracy values and reduce the simulation time to meet the time requirements for real hand classification applications. Chapter 6 will demonstrate the evaluation of self-organising map network as a deep learning suggested model.

CHAPTER 6

Evaluation of Self Organising Map Deep Learning for Biosignals

Feature extraction plays a significant role in biosignal processing. Feature identification and selection is dominated by two common approaches. The standard approach is manual feature extraction, which is based on user experience and application. The other approach is feature learning, which is based on training the model, and identifying and selecting the best features to suit the application. The idea behind feature learning is to avoid dealing with any feature extraction or reduction algorithms and to train the suggested model on learning features from the input biosignal. In this chapter, self-organising maps will be implemented as a feature learning technique to teach the model to extract features from the input data. A deep learning approach is proposed by applying SOMs to learn features. In the proposed model, raw data will be read and then represented using different signal representations such as the spectrogram, wavelet and wavelet packets.

The newly represented data will be introduced to the self-organising map layer to generate features. Then, the performance of the suggested scheme will be evaluated by applying different classifiers such as the support vector machine, extreme learning machine, self-adaptive evolutionally extreme learning machine (SA-EELM) and discriminant analysis classification. Analysis of variance and confidence intervals for different classifiers will be calculated. As an improving step for the results, a classifier fusion layer will be added to select the most accurate result for both the training and testing sets. The classifier fusion layer led to promising training and testing accuracies.

A discussion of the proposed model will be explained in Section 6.2 along with previous work that has been done. The experimental setup along with the results will be presented in Section 6.3. Discussion will be provided in Section 6.4, followed by a summary in Section 6.5.

6.1 Implementation of Self Organising Map Deep learning for biosignal

In this section, the suggested SOM deep learning model will be explained. Figure 6.1 shows the procedures of the recommended system.

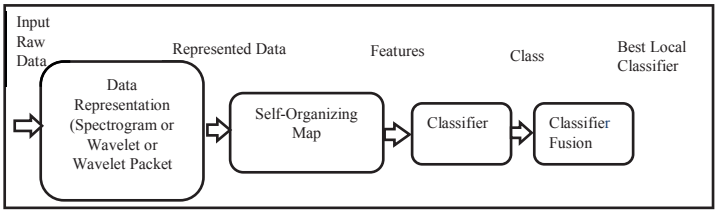


Figure 6. 1: Procedures of suggested SOM feature learning model

In the proposed model, the raw biosignal is represented by the spectrogram, wavelet and wavelet packets individually. This representation stage is very crucial as it gives us the opportunity to present the biosignal better than without representation, which has its impact on the accuracy values. Accordingly, the accuracies become higher than when introducing the biosignal directly to the self-organising layer without applying any signal representation, which leads to accuracy values less than 50%. The output from the signal representation was introduced to the self-organising layer, which is a typical artificial neural network. The output of the SOM is called maps and is of lower dimensional than the input. The self-organising map applies error correction by using the back-propagation technique or gradient descent. The SOM keeps the properties for the input by using a neighbourhood function [295]. Figure 6.2 shows the mapping of a point in the input space to its new point in the output space by using the self-organising map algorithm. The SOM is used to reduce the dimension of the data into lower dimension and more useful representations.

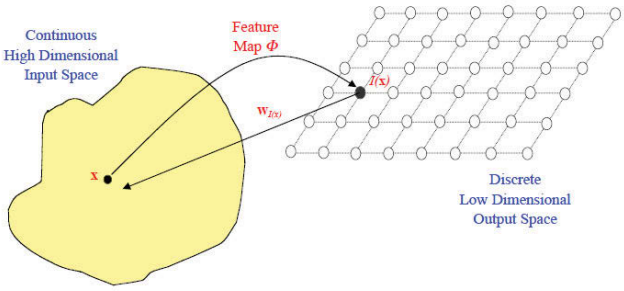


Figure 6. 2: Mapping input space to output space in SOM

The output of the SOM is the new features with a lower dimensional than the input data. Those features are presented to the classifier layer for evaluation. Four classification algorithms are implemented, where the first is the extreme learning machine, the second is the discriminant analysis, the third is the support vector machine, and the fourth is the self-adaptive evolutionary extreme learning machine. The accuracy of six different types of SVM (linear SVM, quad SVM, cubic SVM, fine Gauss SVM, medium Gauss SVM and coarse Gauss SVM) is compared. The accuracy of both linear discriminant analysis and quadrature discriminant analysis is compared and the most accurate result is selected. A brief analysis will be presented on SA-EELM as follows:

The extreme learning machine is used in modelling complex systems, either linear or nonlinear. The modelling algorithm uses a gradient free fast convergence tool [296-299], for N observations $\{(x_i\,, t_i)\}^N$, where$x_i \in R^d$ and $t_i \in R^m$. The ELM chooses both the input weights and hidden biases in a random manner.

$$y_i = \sum_{j=1}^{L} \beta_j \, g\left(a_j . x_i + b_j\right) = t_i \quad , \; j = 1, \dots, N \; , \tag{6.1}$$

where $a_j = [a_{j1}\,, a_{j2}\,, \dots, a_{jn}]^T \in R^d$ is the input weight, x_i is the input signal & $b_j = [a_{j1}\,, a_{j2}\,, \dots, a_{jn}]^T \in R\ (j = 1,2, \dots, L)$ is the bias. Those are the learning parameters of $the\ j$th hidden node, $\beta_j = [\beta_{j1}, \beta_{j2}, \dots, \beta_{jn}]^T \in R^m$ is the output weight that relates the output node and the hidden node. Equation (6.1) can be simplified as.

$$H.\beta = T \quad ,$$

(6.2)

where H is the hidden matrix. Matrix H can be defined as:

$$H = \begin{bmatrix} g(a_1.x_1 + b_1) & \cdots & g(a_L.x_1 + b_L) \\ \vdots & \ddots & \vdots \\ g(a_1.x_N + b_1) & \cdots & g(a_L.x_N + b_L) \end{bmatrix}.$$

(6.3)

The jth element of H represents the corresponding output vector of the jth hidden neuron concerning ith input x_i , the values of this H parameter are selected randomly. Subsequently, the estimate of the β_j^* , a_j^* , and b_j^* parameters will be calculated using equation (6.4).

$$\|H(\, a_1^*, \ldots, a_L^*, b_1^*, \ldots, b_L^*)\beta^* - T\| = \min_\beta \|H(\, a_1, \ldots, a_L, b_1, \ldots, b_L)\beta - T\| \,.$$

(6.4)

The cost function can be defined by the following equation

$$SSE = \sum_{i=1}^{N} \left[\sum_{j=1}^{L} \beta_j \, g\left(a_j.x_i + b_j\right) - t_i \right]^2 ,$$

(6.5)

where SSE is the sum square of error.

The least square solution of the $H.\beta = T$ can be determined as:

$$\beta = H^{\dagger} T \,,$$

(6.6)

Where $H^{\dagger}$ is the Moore-Penrose generalised inverse of the hidden layer matrix H .

The self-adaptive evolutionary extreme learning machine introduces an optimisation algorithm by applying differential evolutionary optimisation techniques to the objective function (θ) . This can be performed by selecting a certain number of populations N_P and a

number of generations G until the targeted level of convergence is obtained [300-304]. The ith parameter can be defined as:

$$\theta_{i,G} = \left[\theta_{i,G}{}^{1}, \theta_{i,G}{}^{2}, \dots, \theta_{i,G}{}^{D}\right] \; i = 1,2, \dots, N_P \;,$$

(6.7)

where D is the described searching space.

The procedures of the differential evolutionary can be defined as follows

First, the cover parameter space should be initialised by using the following equation.

$$\theta_{i,G} = \theta_{min} + rand(0,1).(\theta_{max} - \theta_{min}) \;,$$

(6.8)

where θ_{min}& θ_{max} are the minimum and maximum boundaries respectively. The values of θ_{min}& θ_{max} are predetermined.

Second, the generation of the new mutual vector is estimated by calculating the difference vectors of randomly picked population vector. These can be calculated by one of four strategies.

Strategy 1:

$$v_{i,G} = \theta_{r_1^i,G} + F.(\theta_{r_2^i,G} + \theta_{r_3^i,G}) \;.$$

(6.9)

Strategy 2:

$$v_{i,G} = \theta_{r_1^i,G} + F.\left(\theta_{best,G} - \theta_{r_1^i,G}\right) + F.\left(\theta_{r_2^i,G} - \theta_{r_3^i,G}\right) + \qquad F.(\theta_{r_4^i,G} - \theta_{r_5^i,G}) \;.$$

(6.10)

Strategy 3:

$$v_{i,G} = \theta_{r_1^i,G} + F.\left(\theta_{r_2^i,G} - \theta_{r_3^i,G}\right) + F.(\theta_{r_4^i,G} - \theta_{r_5^i,G}) .$$

(6.11)

Strategy 4:

$$v_{i,G} = \theta_{i,G} + F.\left(\theta_{r_1^i,G} - \theta_{i,G}\right) + F.\left(\theta_{r_2^i,G} - \theta_{r_3^i,G}\right),$$

(6.12)

where $r_1^i, r_2^i, r_3^i, r_4^i$ and r_5^i are mutually exclusive integers numbers which are randomly selected. $0 \leq F \leq 2$ is applied to adjust the scaling difference, and the ruling parameter is randomly created within the range $0 \leq K \leq 1$.

Thirdly, crossover is used to increase the varieties of the confused parameter vectors with respect to a mutant vector $v_{i,G} = [{v_{i,G}}^1, {v_{i,G}}^2, ..., {v_{i,G}}^D]$. At generation G, a trail vector $u_{i,G} = [u^1, {u_{i,G}}^2, ..., {u_{i,G}}^D]$ is generated according to the following equation:

$$u_{i,G} = \begin{cases} {v_{i,G}}^j \; if \; (rand\, j \leq CR \; OR \; (j = jrand) \\ {\theta_{i,G}}^j \; Otherwise \end{cases},$$

(6.13)

Where CR is the crossover rate to adjust the friction of the parameter values copied, rand j is the jth evaluation of a uniform random number created in [0, 1] and jrand is randomly selected an integer from [1, D].

Fourthly, the selection is conducted using fitness function for each objective and corresponding trail vector. The one with the lowest fitness function value is kept as the population for the next generation. Then, the second and fourth steps are repeated until the objective is met or maximum iteration reached.

The self-adaptive evolutionary extreme learning machine is implemented to overcome the restrictions of the mutually selecting trail vector generation strategies. Its associated ruling parameters to optimise the ELM output weight matrix use the following procedures to

initialize the hidden layer matrix

$$\theta_{i,G} = g\left(a^{T}_{1,(k,G)}, \dots, a^{T}_{L,(k,G)}, b^{T}_{1,(k,j)}, \dots, b^{T}_{L,(k,G)} \right),$$

(6.14)

Where $j = 1,2, \dots, L$ are randomly generated, G is the generation, and $k = 1,2, \dots, N_P$ represents the number of population. The output matrix is calculated using the following formula:

$$\beta_{k,G} = H_{k,G}{}^{\dagger} T .$$

(6.15)

$$RMSE_{k,G} = \sqrt{\frac{\sum_{i=1}^{N} \left\| \sum_{j=1}^{L} \beta_j \, g\left(a_{j,(k,G)} . x_i + b_{j,(k,G)}\right) - t_i \right\|^2}{m.N}},$$

(6.16)

where, $H_{k,G}$ and $\theta_{k,G+1}$ can be defined by using the following equations

$$H_{k,G} = \begin{bmatrix} g\left(a_{1,(k,G)} . x_1 + b_{1,(k,G)}\right) & \cdots & g\left(a_{L,(k,G)} . x_1 + b_{L,(k,G)}\right) \\ \vdots & \ddots & \vdots \\ g\left(a_{1,(k,G)} . x_N + b_{1,(k,G)}\right) & \cdots & g\left(a_{L,(k,G)} . x_N + b_{L,(k,G)}\right) \end{bmatrix} .$$

(6.17)

$$\theta_{k,G+1} = \begin{cases} u_{k,G+1} \; if \; RMSE_{\theta k,G} - RMSE_{uk,G+1} > \varepsilon . RMSE_{\theta k,G}, \\ u_{k,G+1} \; if \; \left| RMSE_{\theta k,G} - RMSE_{u\,k,G+1} \right| < \varepsilon . RMSE_{\theta k,G} \\ \qquad\qquad and \; \left\| \beta_{uk,G+1} \right\| < \left\| \beta_{\theta k,G} \right\| \\ \theta_{k,G} \quad elsewhere \end{cases} .$$

(6.18)

All trail vectors $u_{k,G+1}$ created during the $(G+1)^{th}$ generation are evaluated by implementing equation (6.16) with small a tolerance ε.

The output from the classifiers was introduced to the classifier fusion layer where the results were improved significantly as will be discussed in the results section.

Analysis of variance was calculated for different classifiers. The average testing accuracies for the three signal representation techniques were gathered (spectrogram, wavelet and wavelet packet) and resulted in P value equal to 0.0613. The P value indicated that there was no practical difference between any of the four classifiers. The confidence interval for each classifier was calculated with 60% confidence score. The confidence level is 60 % as long as the testing accuracy, for each classifier, within the allocated confidence interval.

6.1.1 Previous work

In this chapter, a deep learning model that will be able to learn features without applying any feature extraction or reduction technique will be suggested. The proposed model will be skilful in classifying between ten-finger movements. The finger movement classification was presented previously in many types of research as in [305] where the researchers proposed a feature projection technique by implementing the fuzzy neighbourhood preserving analysis (FNPA) with QR-decomposition. The primary purpose of FNPA is to decrease the distance, to the maximum extent, between the samples of the same class and increase it between samples of different classes. The researchers' purpose was to classify ten-finger movements, and they achieved an average accuracy 91% by using two channel electrodes. After that, other authors proposed a finger movement classification system. Where, they employed the spectral regression discriminant analysis for dimensionality reduction. The extreme learning machine for classification and smoothed the results by using the majority voting method. They achieved 98% classification accuracy from two electromyography channels [306]. Then the authors used the same data set to apply the spectral regression discriminant analysis for dimensionality reduction. The kernel-based extreme learning machine for classification and smoothed the results by using the majority voting method [307]. Moreover, they applied the particle swarm optimisation to optimise the

kernel based ELM. Later, other researchers suggested a model classifying between nine-finger movements by using two electrodes [308]. They used seven-time domain features. And evaluated the system by using the artificial neural network and k-nearest neighbours as classifiers. Other researchers suggested using the nonnegative matrix factorization (NMF) technique to select the accurate and dependable features. They employed the neural networks for classification. And calculated the accuracy for both simple and complex flexion. Where they achieved 95% for simple flexion and 87% for complex one [309]. After that, other researchers suggested collecting the surface electromyography signals using eight electrodes to classify between nine hand motions. They used two different neural networks for the classification. The first neural network architecture attained 83.43% accuracy whereas the second one achieved 91.85% [310]. Another experiment [311] where the authors applied time-domain descriptors (TDD) to measure the power spectrum characteristics for the electromyography signal. This reduced the computationally expensive feature construction traditional techniques. This suggested approach improved the error percentage by 8% in comparison with other approaches [311]. Another trial was published for recognising finger movements by using the microneedle with 94.9% average accuracy [312]. On the other hand, many researchers claimed for applying a general model that should be able to learn features. This approach learns features without employing any feature extraction and reduction techniques, which led to the deep learning clue. Many published experiments were done in using the deep learning for the biomedical data. As for example but not limited to, [313] the authors developed a system by using an ensemble of the convolutional neural networks to classify medical images. The convolutional neural networks were implemented for learning features by tunning and for the classification. Other researchers suggested employing the convolutional neural networks to learn features and developing low-level features only [314]. The authors [315] developed a deep learning system to classify between different plaque constituents in carotid ultrasound. Later, the texture of lung pattern was analysed by implementing the deep learning

technique after employing the convolutional neural networks [316]. Consequently, the deep learning can be considered as a feature learning system where the system extracts the best features suit the application. In other words, the system learns features without employing any of the feature extraction or feature reduction algorithms. A deep learning model will be proposed via the self-organising map in learning features from the represented biosignals. The biosignals will be represented using either the spectrogram, wavelet or wavelet packets. The behaviour of the suggested model will be evaluated by using different classifiers as the support vector machine, discriminant analysis, extreme learning machine and evolutionally extreme learning machine. The evolutionally extreme learning machine will lead to higher accuracy results than other implemented classifiers. The analysis of variance and confidence intervals for different classifiers will be calculated. Finally, and as a development stage, a classifier fusion layer will be added to select the best local classifier. Accordingly, the proposed model will learn features without applying any feature extraction or reduction algorithms and achieve high accuracy values. Moreover, implementing the analysis of variance and confidence intervals is considered as an in-depth step to understand the nature of the relation between the employed classifiers and to know the trustful range of the accurate results.

6.2 Experiments and Results

In this section, the SOM layer settings, classifiers and classifier fusion layer will be presented. The simulation outcomes will be presented and discussed. The data acquisition, filtering, amplification, and experimental setup were identical to that explained in Chapter 4.

6.2.1 Results

In this proposed model, the biosignals were represented by either the spectrogram, wavelet or wavelet packets before being introduced to the SOM layer. Feeding the SOM with raw biosignals directly, without applying any further representation led to

accuracy values less than 50%. Four classifiers were implemented, which were ELM, SA-EELM, DA and SVM. SA-EELM achieved a better classification ability than the other three classifiers. Applying the classifier fusion algorithm led to a substantial improvement in the accuracy values. The ANOVA and confidence intervals were estimated. Table 6.1 shows the training and testing accuracies for the different signal representations and different classifiers.

Table 6. 1: SOM deep learning training and testing accuracy

Signal Representation	Average Training Accuracy	Average Testing Accuracy	Classification Algorithm
Spectrogram	99.41%	81.54%	Support Vector Machine
Spectrogram	98.26%	83.29%	Extreme Learning Machine
Spectrogram	96.71%	91.11%	Evolutionally Extreme Learning Machine
Spectrogram	96.95%	83.13%	Discriminant Analysis
Wavelet	99.84%	85.11%	Support Vector Machine
Wavelet	98.03%	84.91%	Extreme Learning Machine
Wavelet	99.35%	93.14%	Evolutionally Extreme Learning Machine
Wavelet	98.61%	86.77%	Discriminant Analysis
Wavelet Packet	98.42%	90.47%	Support Vector Machine
Wavelet Packet	97.83%	87.96%	Extreme Learning Machine
Wavelet Packet	99.50%	94.44%	Evolutionally Extreme Learning Machine
Wavelet Packet	96.30%	89.98%	Discriminant Analysis

Different experiments were performed by altering the SOM configuration until the current configuration was reached where the size of the two-dimensional map was 10x10, the layer topology function employed was ‘hextop’ and the neurone distance function used was ‘linkdist’. This configuration was to compromise between the simulation time and accuracy level. These parameters were optimum with regard to accuracy level and the time spent in the simulation. Increasing the size of the map would consume more simulation time without a tangible impact on the accuracy level. The biosignals were filtered before the signal representation stage and the neural

networks phase. This was to guarantee a substantial barrier against the noise level that might be superimposed on the collected biosignals.

The same set of data were introduced to a pattern recognition system, as shown in Figure 6.3. Traditional feature extraction and feature reduction techniques were applied. The extracted features were slope sign change, zero crossing, waveform length, skewness, mean absolute value, root mean square and autoregressive. Linear discriminant analysis was used for feature reduction. ELM was used for the classification task, where the training accuracy was 96.60% and the testing accuracy was 88.92%.

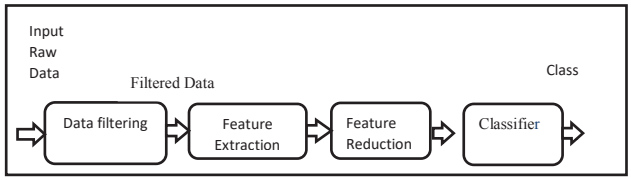


Figure 6. 3: Pattern Recognition Model

It is clear from the results that are shown in Table 6.1 that using the wavelet packets as a signal representation generated a higher testing accuracy than both the wavelet and spectrogram. Moreover, the wavelet had a higher testing accuracy than the spectrogram and had a lower testing accuracy than the wavelet packets. The support vector machine achieved better classification ability for the wavelet packets than its ability in the spectrogram. The support vector machine testing accuracy for the wavelet packets was very close to the testing accuracy for the discriminant analysis classifier but higher. Moreover, the support vector machine attained a moderate classification ability for the wavelet and poor classification ability for the spectrogram. On the other hand, the extreme learning machine exceeded the other classification techniques for the spectrogram. However, testing accuracy for the spectrogram was very close to its value, resulting from using the discriminant analysis classifier. The discriminant analysis achieved a higher testing accuracy for the wavelet than both the extreme learning machine and support vector machine. Testing accuracy for the wavelet using the

support vector machine, was higher than the testing accuracy resulting from the extreme learning machine. The testing accuracies, for the three signal representations, were boosted to another higher level by implementing the evolutionally extreme learning machine as a classifier. This is due to the optimisation algorithm that is used by this technique. The only drawback for the evolutionally extreme learning machine was its simulation time. It was considered longer than the time spent during the simulation of the extreme learning machine. However, the high accuracy levels gained from the evolutionally extreme learning machine were encouraging.

The ANOVA value for the four different classifiers was calculated. The resulting P value was 0.0613, which is slightly higher than 0.05. However, it indicated that there was no practical difference between those four implemented classifiers.

Table 6.2 shows the confidence intervals for the four different performed classifiers.

Table 6. 2: Confidence interval for various classifiers

Classifier	Confidence Interval High	Confidence Interval Low	Confidence Interval
Extreme Learning Machine	86.54%	84.24%	2.3%
Discriminant Analysis	88.29%	84.96%	3.32%
Support Vector Machine	87.89%	83.53%	4.36%
Evolutionally Extreme Learning Machine	93.71%	92.08%	1.63%

Figure 6.4 shows the clustered column chart, which represents the confidence interval for the applied classifiers. The implemented confidence score was 60%. It was apparent from these results that the evolutionally extreme learning machine had the narrowest range which was 1.63%. The support vector machine had the broadest interval, which influenced the gap between higher and lower confidence intervals. The second most extensive interval was the discriminant analysis that resulted in 3.32%. Finally, the extreme learning machine generated the second narrowest interval (2.3%).

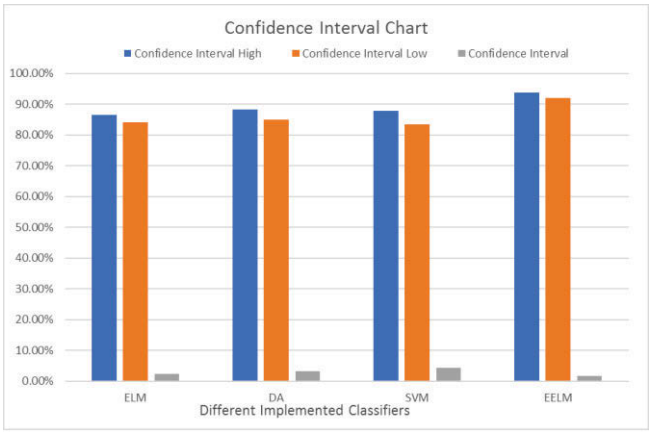


Figure 6. 4: Confidence interval for four classifiers

A classifier fusion layer was added where it selected the best local classifier by using the dynamic classifier selection algorithm. Both the training and testing accuracies were improved as shown in Table 6.3.

Table 6. 3: Classifier fusion training and testing accuracy

Signal Representation	Average Training Accuracy	Average Testing Accuracy	Classification Algorithm
Spectrogram	98.42%	93.07%	Classifier Fusion
Wavelet	99.35%	95.25%	Classifier Fusion
Wavelet Packet	99.73%	96.60%	Classifier Fusion

The testing accuracies were improved by adding classifier fusion where the wavelet packets testing accuracy exceeded 96%. And the wavelet testing accuracy became higher than 95%. The spectrogram increased to more than 93%. In comparing Table 6.3 results with Table 6.1 results, it became clear that classifier fusion improved the accuracies.

6.2.2 Comparison between the proposed model with other researchers' achievements

In this section, we will compare between the proposed model and other researchers' achievements for the same dataset. The authors [74] implemented the same two-

channel data set that was utilised in the SOM. The researchers suggested a pattern recognition model, used traditional feature extraction, and used SRDA for feature reduction. The extreme learning machine was employed for the classification and majority vote for smoothing. The researchers classified the same ten-finger gestures. The authors applied time domain features such as sample skewness, slope sign changes, number of zero crossings and waveform length. Autoregressive and Hjorth time domain parameters were also estimated. Furthermore, SRDA was employed for feature reduction rather than linear discriminant analysis. It was found that SRDA achieved faster performance than LDA. This made SRDA more suitable to deal with more significant number of features than LDA. Additionally, and as a comparative study, the authors applied regularised LDA (RLDA) for feature reduction beside SRDA. Regards classification, the researchers found that ELM achieved high accuracy values and showed superior classification performance, similar to that of the support vector machine with less time. Table 6.4 shows the average testing accuracy and the average simulation time for the pattern recognition model presented.

Table 6. 4: Average accuracy and simulation time for SRDA and RLDA

AR order	Average Classification Accuracy (%)		Average Time (ms)	
	RLDA	SRDA	RLDA	SRDA
2	89.89 ±8.56	91.23 ±7.56	1.40 ±0.27	1.30 ±0.37
4	94.70 ±4.53	93.68 ±6.27	2.30 ±0.41	1.60 ±0.37
8	94.50 ±5.56	94.66 ±5.47	3.00 ±0.09	1.80 ±0.42
16	95.73 ±5.08	96.45 ±4.69	4.50 ±0.46	2.60 ±1.10
32	97.20 ±3.66	97.57 ±2.78	8.10 ±0.62	3.60 ±0.69
64	97.48 ±2.94	97.71 ±3.44	19.80 ±1.40	7.30 ±0.98
100	98.48 ±2.17	98.45 ±2.64	41.20 ±3.30	14.30 ±3.30

As it is clear from Table 6.4, testing accuracy was improved as the autoregressive order increased until achieving the best performance for an autoregressive order of 100. RLDA and SRDA showed similar performance in testing accuracy as the autoregressive order increased. SRDA spent less than that spent by RLDA especially for higher autoregressive order. The highest accuracy was 98.45 ±2.64 % for autoregressive order 100. On the other hand, the proposed deep learning SOM achieved

96.60% testing accuracy and 99.73% for the training set. Furthermore, the SOM scheme was able to learn the features by itself without recourse to traditional feature extraction, and reduction algorithms. The traditional feature extraction and reduction approaches are time-consuming and dependent on the user's experience along with the application. However, the recommended deep learning model did not acquire any human intervention in regards to the features as the system was capable of learning features that best match the application. Nowadays, deep learning models have attracted many researchers in different fields as they reduce the burden of pattern recognition users in extracting and reducing the best features that match the application.

6.3 Discussion

SOM is a timesaving and easy feature learning approach. It reduced the dimensions of the data without too much loss in information and observations that should be associated with the data. The self-organising map is a neural network that uses the dimensionality reduction (typically two dimensions) and backpropagation for the error correction. Although the simulation time was not long, good testing accuracies were obtained.

The spectrogram, wavelet and wavelet packets helped in better representing the raw biosignals. The support vector machine achieved good classification ability in the wavelet packets, and discriminant analysis attained an impressive result for the wavelet. The extreme learning machine was superior in the spectrogram testing accuracy to the discriminant analysis.

The evolutionally extreme learning machine achieved a genuine improvement for the accuracy percentages. This result was due to the optimisation algorithm that existed in the model, as all testing accuracies, for evolutionally extreme learning machine, exceeded 90%. The only drawback for the evolutionally extreme learning machine was its simulation time as it was relatively longer than that consumed during implementing the extreme learning machine. On the other hand, the accuracy values achieved by employing the evolutionally extreme learning machine as a classifier was higher than that attained by using the extreme learning machine.

The ANOVA value was 0.0613 as calculated above. The value was not yet to be less than 0.05. However, it was very close to 0.05. This P value meant that there was no practical difference between the classifiers. Despite this, it was believed that using the evolutionally extreme learning machine pushed the ANOVA value to be as close as 0.05.

The confidence intervals were calculated by a confidence score of 60%. As shown above, the narrowest interval was for the evolutionally extreme learning machine. The broadest range was for the support vector machine. These intervals gave us an indication of how confident we were in our resulting testing accuracy. In addition, if the testing accuracy was in the confidence interval range, that meant that the result is trustworthy by 60%.

The classifier fusion layer enhanced testing accuracies for the three used signal representations. The classifier fusion algorithm that followed selected the best local classifier. The only drawback for classifier fusion was increasing the simulation time as it became relatively high. However, the simulation time consumed by the self-organising map as feature learning was short.

The average simulation time was around 20 seconds, which was considered to be a short time. However, it was far from a real-time hand application (150ms to 300 ms).

6.4 Summary

The use of self-organising maps for deep learning was discussed in this chapter. However, the biosignals had been represented before being introduced to the self-organising map layer. The implemented signal representations were the spectrogram, wavelet and wavelet packets. Feeding the biosignals directly to the self-organising map layer, without applying any signal representations, brought an accuracy value less than 50%. The extreme learning machine, self-adaptive extreme learning machine, support vector machine and discriminant analysis classifiers were used where SA-EELM achieved superior performance to the other executed classifiers. This outstanding behaviour was due to the existing optimisation algorithm that was embedded in the SA-EELM classifier. The classifier fusion layer was used to boost the accuracies for higher levels and achieved accuracy higher than 96%. The ANOVA was

estimated by P value 0.0613. This value is higher than 0.05, however, it was close to it. As the SA-EELM caused a boost in the accuracy values. This pushed the ANOVA value to be close to 0.05. The confidence intervals were assessed at the 60% confidence level. This interval could be inferred as long as the accuracy result fell in the deduced interval. The trustworthness of this result reached the level of 60%. The simulation time was not long compared to the autoencoder implementation for the deep learning mentioned in chapter 4. However, the simulation time is still far from a real-time hand application, which should consume 150 ms to 300 ms. The applied SOM deep learning scheme showed high accuracy values for both the training and testing set along with consuming short simulation time. The generalisation was one of the key objectives of this thesis that inspired the motivation of executing the data augmented model. Accordingly, a data augmented model will be explained in Chapter 7. This model will be applied to different datasets and for various applications. This made the model gain more fidelity and trust. Chapter 7 will present the execution of data augmentation along with a tensor representation for the deep learning implementation.

CHAPTER 7

Evaluation of Augmented Data with Tensor Representation for Deep Learning for Biosignals

Data representation is one of the most motivating aspects to the accuracy of our deep learning model. Nowadays, deep learning technology imposes its presence in the field of feature learning to save time and effort for researchers, as the typical feature extraction techniques depend on user experience in selecting the most powerful features. On the other hand, deep learning creates a model that is capable of learning the features without any external intervention. In this chapter, the deep learning concept will be applied via the implementation of the autoencoder. The main contribution of this chapter will be the approaches of representing the biosignals. This will provide us the opportunity to examine different biosignals data sets, on the same proposed model, and achieve high accurate results. The spectrogram of the biosignals will be estimated. and Gaussian representation for the spectrogram will be implemented as a data augmentation (DA) phase. This augmented set will be represented in the tensor domain which will be introduced to the two stages autoencoder to learn features. Ultimately, three different classifiers will be used. These classifiers are the support vector machine, extreme learning machine, and Softmax layer (SL). A classifier fusion layer will be implemented to select the best local classifier. The results of a typical feature extraction model will be compared with the suggested deep learning model for benchmarking. Furthermore, different data sets for district applications will be introduced to the proposed model to claim generalisation. It can be alleged that the suggested model is capable of learning features from the biosignals for different applications. A discussion of the proposed model will be presented in Section 7.2. and introduce previous work on the topic. The experimental setup along with the results will be explained in Section 7.3. A Discussion will be provided in Section 7.4, followed by a summary in Section 7.5.

7.1 Implementation of augmented data with tensor representation for deep learning for biosignals

In this section, the suggested augmented data with tensor representation for the deep learning model will be explained. Figure 7.1 shows the procedures of the recommended system.

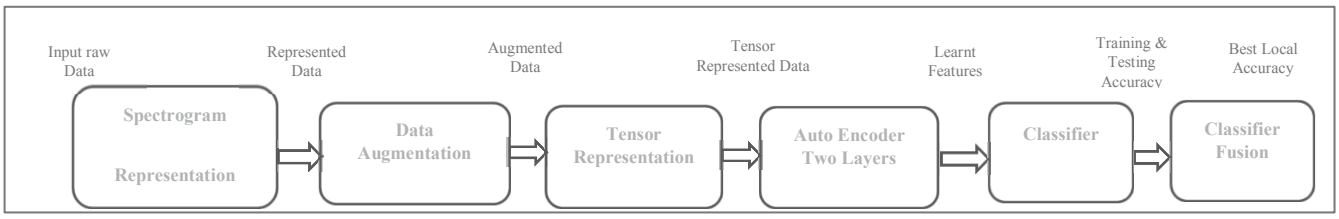


Figure 7. 1: Procedures for data augmented tensor representation deep learning model

It is claimed that the suggested paradigm is able to achieve high accuracy values for both the training and testing data sets with a powerful signal representation. As a preparatory step, the input raw biosignals were represented by a spectrogram. The implemented window size is 200 ms and the window increment is 50 ms. The recommended number of sampling points to calculate the discrete Fourier transform is 1024. The advantage of proceeding with the spectrogram is that it provides a reliable representation for the raw input biosignals. This in turn boosted the accuracy of the values for both the training and testing set. The spectrogram output is introduced to the DA layer, and the above-mentioned Gaussian augmentation is used by repeating the represented data 1000 times. DA enriches and grants affluence to the data and, in return, improves the training and testing accuracies. At the final stage of data representation, tensor representation is used to represent the data in a developed perspective. The data are then introduced to the two layers autoencoder to learn features from the high quality represented data. The first layer of the autoencoder is 1200 nodes and the second layer is 900 nodes.

The weight regularisation coefficient is set to 0.001, as it is the default value, for both layers of the autoencoder. The coefficient that controls the weights of the sparsity regularisation is set to 4 for both layers. The sparsity proportion factor determines the activation response rate of the autoencoder neuron. The value of the sparsity proportion varies between 0 to 1, where lower value promotes and inspires for a higher sparsity. The sparsity proportion is set to 0.05. The

encoder transfer function is set to pure linear. Additional transfer functions are used, such as the logistic sigmoid and positive saturating linear transfer functions. However, the linear transfer function leads to the highest results. The output learnt features are employed to proceed to the classifier phase. Three different primary classifiers are used. The utilised classifiers are the ELM, SVM and SL. For the ELM, five activation functions are used, and we select the activation function that generates the most precise results. The five executed activation functions are the sigmoidal, sine, hardlim, triangular basis and radial basis functions. As for the SVM classifier, we examine the performance of six different functions for the SVM. The executed SVM functions are the linear, quadratic, cubic, fine Gaussian, medium Gaussian and coarse Gaussian SVM again we selected the SVM function that leads to the best outcome. The accuracies of the three classifiers are presented to the classifier fusion layer to select the best local classifier per class.

7.1.1 Previous work

DA has been widely used and implemented in many research. The researchers established a relationship between the iterative computation time for the expectation maximisation procedure and the extension of the space parameter by augmenting the data [317]. The recognised relation was an expansion of the space parameter, which was applied with the DA. The iterative computation time was expected to be shorter. Earlier, scientists calculated the posterior probability for augmented data when the average likelihood could not be reached [318]. DA has been applied in different fields, for example replication data in the field of chemistry [319]. Image recognition was a state of the art field that the DA was innovatively applied. The researchers used manual augmentation techniques in conjunction with deep neural which led to an enhanced achievement [320]. Moreover, the researchers implemented the DA algorithm for a hand-drawn draughts dataset. They used a fine-tuned deep convolutional neural network to extract useful features from the introduced dataset [321]. Recently, the authors recruited the data augmentation Markov chain Monte Carlo (MCMC) [322].

Furthermore, scholars have applied the augmentation in both the data domain and feature domain [323]. The convolutional neural network was applied for an acoustic signal, whereas the authors applied the augmentation to the speech signal to prove that the gap between the real room impulse response and simulated one was reduced to its minimum value [324]. In addition, the authors combined deep belief networks with the data augmentation algorithm, which added gamma variables to the original signals [325]. In the field of image processing, the authors augmented the input image by generating a 3D copy to be processed in a convolutional neural network [326]. Finally, the researchers applied depuration in addition to augmentation and balancing to Electroencephalography (EEG) signals [327].

Tensor representation was used in different research fields to allow a better representation of a dataset. In the early of 20^{th}-century, the scientists paid attention to the value of the tensors and its applicable fields [328]. In the field of continuum mechanics, the tensor fundamentals and enforcement were discussed [329]. A paper addressed tensor decomposition technique and treated it as a generalisation for matrix decay [330]. In the concept of the deep tensor neural network (DTNN) was also initially introduced, where a double projection layer substituted one of the convolutional layers [331]. Those two inserted subspaces are entirely nonlinear, therefore any input of vocabulary speech was mapped to the newly introduced subspaces in conjunction with a tensor layer. This model was capable of anticipating the next layer in the deep neural network design. Also, this proposed model resulted in a relatively reduced error by 3%. A model that was able to estimate an approximation for the tensor rank has been developed [330, 332]. One by disintegrating the tensor and estimate canonical polyadic decomposition (CPD) by using a sparse matrix of banded type. And the other one, in 2014, by the inequality of the M tensors was discussed and resulted in the upper and lower values for the eigenvalues [333]. The authors and resulted in the upper and lower values for the eigenvalues [334]. The estimation of the tensor parameters was analysed

deeply [335]. Furthermore, the tensor decomposition techniques were presented in [336] to give the opportunity for more latent dataset than that based on matrix domain. Recently, the tensor decomposition was applied on the genetic expression to a group of a latent component [337]. This latent component was used to find a relation between any biological development and a genetic variation.

7.2 Experiments and Results

In this section, the data acquisition methods we followed will be expressed more extensively. The experimental setup, including biosignals filtering and amplification will be revealed. The different datasets will also be shown along with the autoencoder layers' settings, classifiers and classifier fusion layer. Additionally, simulation outcomes will be shown and discussed.

7.2.1 Data Acquisition

Different data sets were inputted into the proposed model for generalisation. The first data set was that used in Chapter 4. This data set was of nine healthy patients to classify ten-finger movements. The second data set was of amputee patients missing their left hand. The objective of collecting this data were to classify between six gestures to understand and analyse the electromyography controlled upper limbs prostheses. The six gestures were thumb flexion, index flexion, fine pinch, tripod grip, hook grip and spherical grip. It was challenging task record the surface electromyography signals from amputee participants with three different force level. The skin was cleaned with alcohol and prepared with the abrasive method. Ag/AgCl electrodes were used.

The surface electromyography signals were recorded by eight channels at three levels of force for nine amputee participants. The first dataset was amplified by a gain of 1000 and the first and second datasets were sampled at 2000 Hz. Figure7.2 shows the allocation of electrodes and collection of the surface electromyography signals from amputee participants.

Figure 7. 2: Electrodes allocation for amputee participants [87]

For the datasets mentioned above, the three-fold cross-validation algorithm was applied where two out of three of the dataset was assigned to the training set and the remaining third was allocated to the testing set. The data were filtered to secure the precision and removal of noise. The training and testing accuracies were estimated on an average basis. The accuracy was calculated per subject whereas the overall accuracy was the summation of the accuracy per subject divided by the number of subjects.

The other datasets were imported from the University of California, Irvine (UCI) machine learning repository "https://archive.ics.uci.edu/ml/index.php". The UCI website is considered a substantial archive and has been cited for more than 1000 times by the machine learning community by either students or researchers. The performance of the proposed model was examined with an additional five datasets that were archived at UCI machine learning repository website. Those datasets were the 'iris', 'breast cancer', 'seeds', 'sonar', 'mines vs rocks' and 'Indian liver' patient dataset. The 'iris' dataset is one of the most popular data set in pattern recognition field. This data set contains three classes. One class is linearly independent of the other two classes and can be easily separated, whereas the other two classes were not separable. The three targeted classes for the 'iris' dataset was iris setosa, iris versicolour and iris virginica. The breast cancer data set was collected from the Dr Wolberg clinic and classified for six classes two of these were benign, Where the other four classes were dedicated to a malignant type. The seeds dataset was collected for three different sorts of wheat. These

three classes of seeds dataset are Kama, Rosa and Canadian, which are recorded via X-Ray plates. The sonar, mines vs rocks data set are used to classify between metal and rock. The last dataset was the Indian liver patients dataset. The dataset contained information collected from 441 male and 142 female Indian participants to discriminant whether or not they are a liver patient.

7.2.2 Results

In this section, the results will be displayed and discussed for all implemented data sets. The following results were estimated after classifier fusion stage, as the implementation of a classifier fusion layer increased the accuracies for both the training and testing set. The classification of the ten-finger movements dataset was 100% for the training pack and 90.25% for the testing set. As for the high force six-finger movements dataset, the training collection accuracy was 99.74%, whereas the accuracy for the testing group was 91.85%. The deep neural network was replaced by a typical pattern recognition model where the features were extracted and reduced by using linear discriminant analysis. The ELM was used as a classifier. Based on the used pattern recognition model, the training accuracy for the ten-finger movements was 95.76% and the testing accuracy was 87.11% as shown in Figure 7.3. As for the six-finger movements, both the training and testing accuracies were lower than those values achieved by the proposed model. The training accuracy was 98.57%, and testing one was 89.64% as shown in Figure 7.4.

This study concluded that the behaviour of the proposed model was explicitly better than the typical pattern recognition model. Furthermore, the suggested system did not require any feature extraction, as it was trained to learn features independent of the input data. Accordingly, the planned scheme was examined on widely used datasets to provide reliability and trust in the model. The implementation of the iris data set resulted in 100 % training accuracy and 98.5%testing accuracy. The breast cancer tissue dataset achieved 98.58% training accuracy and 91.7% testing accuracy. For the

sonar dataset, the accuracy of the training set was 85.69%, and of the testing set was 74.4%. Moreover, examining the liver dataset led to 96.47% training accuracy and 85.1% for testing accuracy. For the seeds data, the training accuracy was 94.57% whereas 83.6% it was for the testing set. The UCI machine learning repository datasets were executed without using classifier fusion layer and only used the softmax classifier. The training simulation time was more than 600 seconds. However, the time consumed for the testing set on the trained network was not more than 1.5 seconds. Table 7.1 shows both the training and testing accuracies for all the above-mentioned datasets. Figure 7.3 shows a comparison between the testing and training accuracies for the suggested model, and those resulting from implementing a typical pattern recognition technique for classifying ten-finger movements. Figure 7.4 displays the same comparison for the six-finger movements. The recommended model not only achieved a better performance on the level of the training and testing accuracies, it also saved the effort and time that might be wasted in selecting the best features that match the application.

Table 7. 1: Training and Testing accuracies for all implemented datasets

Dataset	Training Accuracy	Testing Accuracy
Ten Finger Movement (Suggested Deep Learning Model)	100%	90.25%
Six Finger Movement (Suggested Deep Learning Model)	99.74%	91.85%.
Ten Finger Movement (Typical Pattern Recognition)	95.76%	87.11%
Six Finger Movement (Typical Pattern Recognition)	98.57%	89.64%
IRIS (Suggested Deep Learning Model)	100 %	98.5%
Breast Tissue (Suggested Deep Learning Model)	98.58%	91.7%
Sonar (Suggested Deep Learning Model)	85.69%	74.4%
Seeds (Suggested Deep Learning Model)	94.57%	83.6%
Liver (Suggested Deep Learning Model)	96.47%	85.1%

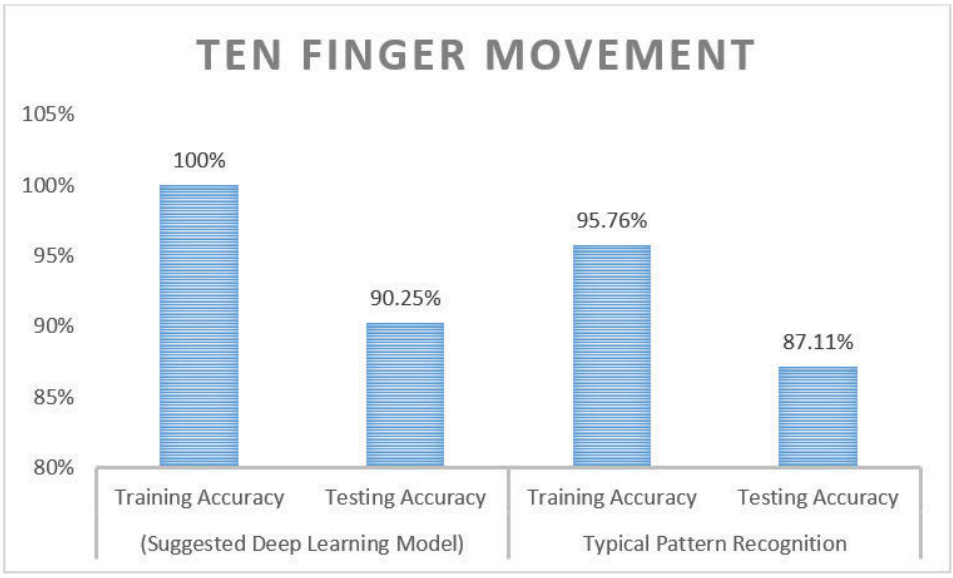


Figure 7. 3: Comparison between suggested deep learning model and pattern recognition in ten finger movement.

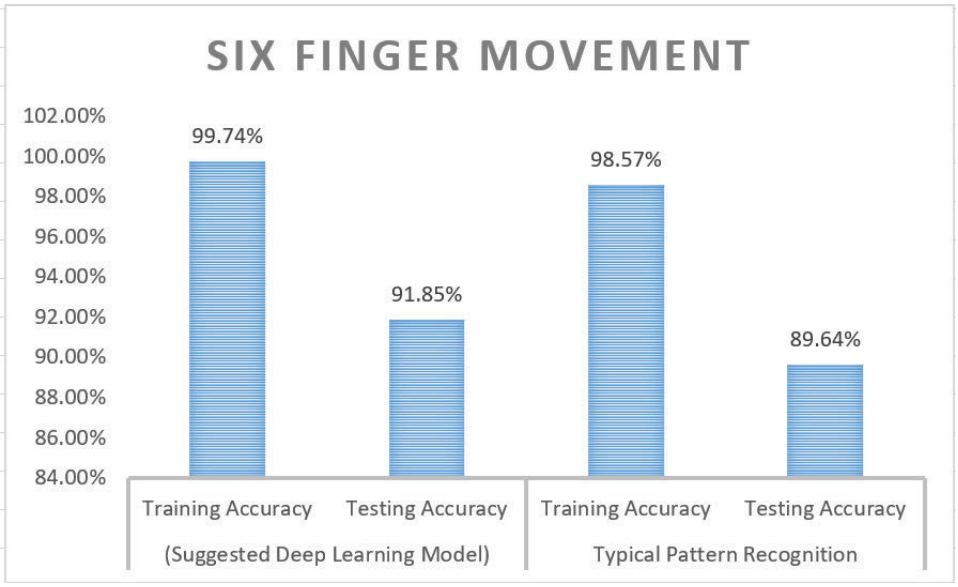


Figure 7. 4: Comparison between suggested deep learning model and pattern recognition in six-finger movement.

7.2.3 Comparison between the proposed model with other researchers' achievements

In this section, the proposed model will be comapred with other researchers achievements for the same dataset. The ten-finger movement dataset that was implemented in the proposed data augmentation deep learning scheme was used by researchers where the authors presented a traditional pattern recognition model for classifying the ten-finger movements [80]. The suggested pattern recognition system adopted a new feature reduction algorithm that merged the ELM with spectral regression (SR) to be SRELM, which was employed for dimension reduction. The authors recorded the surface electromyography signals for the ten-finger gestures, and extracted time-domain features other than the autoregressive for every 100 ms time frame. The extracted time-domain features were the number of zero crossings, zero waveform length, mean absolute value, sample skewness and slope sign changes. The sixth order autoregressive and parameters from the Hjorth time-domain features were also estimated. SRELM was used for the dimension reduction purposes. The number of features were reduced to the number of classes. The authors compared the SRELM performance with that of the uncorrelated linear discriminant analysis, orthogonal fuzzy discriminant analysis and spectral regression dimensionality reduction. Moreover, the researchers recruited the unsupervised extreme learning machine, principal component analysis and baseline study where no dimension reduction algorithm was applied. The authors applied five classifiers: the linear discriminant analysis, support vector machine, radial basis function ELM, adaptive wavelet ELM and k-nearest neighbourhood. The authors simulated seven dimension reduction algorithms including no feature reduction (baseline) techniques and classified the gestures via AW-ELM. Moreover, the researchers implemented five to ten finger movements in their study. The ten-finger movements' results were the principal concern as it was to be compared with the ten finger gestures' outcomes of the suggested data augmented deep learning scheme. The targeted ten-finger gestures were

thumb (T), index (I), middle (M), ring (R), little (L), thumb-index (T–I), thumb-middle (T–M), thumb-ring (T–R), thumb-little (T–L) and the hand close (HC). The first five gestures were single finger movements whereas the last five were combined actions. Table 7.2 shows the different classes combinations used in the study.

Table 7. 2: Different classes' combinations were implicated in the study

#Classes	Classes
5	T, I, M, R, L
6	T, I, M, R, L, T-I
7	T, I, M, R, L,T-I, T-M
8	T, I, M, R, L, T-I, T-M, T-R
9	T, I, M, R, L, T-I, T-M, T-R, T-L
10	T, I, M, R, L, T-I, T-M, T-R, T-L, HC

Figure 7.5 shows the average accuracy of the seven dimension reduction algorithms including no feature reduction (baseline) technique grouped by the various class combinations for the AW-ELM classifier only.

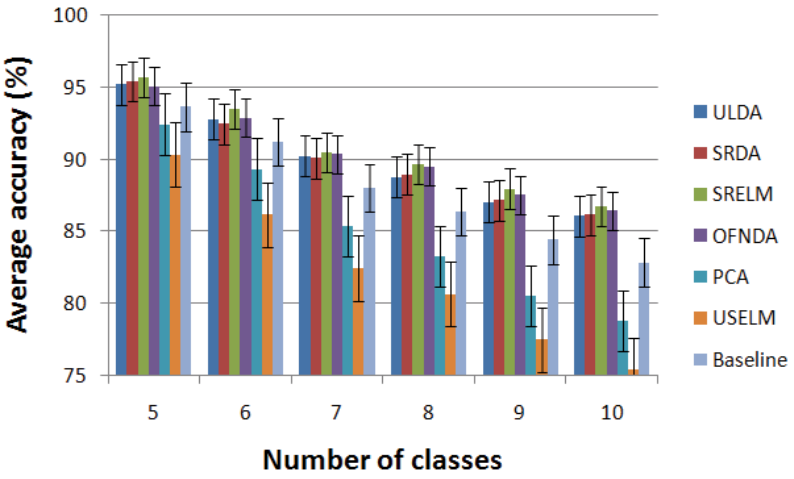


Figure 7. 5: Average accuracy grouped by the number of classes for various dimension reduction [80]

The SRELM attained a superior performance to the other implemented feature reduction techniques. In addition, the average accuracy decreases as the number of targeted classes increases. The accuracy values achieved 95.67% for five classes and reached 86.73 % for ten classes. Afterwards, the authors examined the performance of the ten-finger gestures with all dimension reduction algorithms except for the

unsupervised extreme learning machine. The five executed classifiers were the LDA, SVM, RBF-ELM, AW-ELM and KNN. Figure 7.6 shows the average accuracy grouped by the five implemented classifiers. The RBF-ELM classifier with the SRELM feature reduction achieved 86.73% highest accuracy.

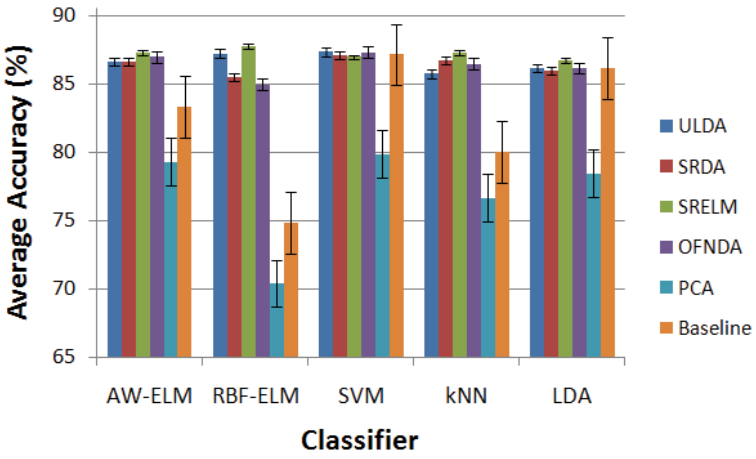


Figure 7. 6: Average accuracy grouped by five classes for different dimension reduction algorithms [80]

On the other hand, the data augmentation deep learning model achieved accuracy 90.25% for the ten-finger movements. Furthermore, the performance of the data augmentation deep learning model was examined on various datasets with different applications. As six-finger movements for amputee participants that attained 91.85% accuracy. The datasets mentioned above were introduced to a traditional pattern recognition model, and achieved lower accuracy values than those attained by the recommended deep learning model. Moreover, the Iris, breast cancer, sonar, seeds and liver patients' datasets were introduced into the suggested deep learning model, and achieved high accuracy values without applying any feature extraction or reduction algorithms. The recommended deep learning model showed a superior performance for different datasets, which provides evidence for the fidelity, generalisation and the trust in the scheme. The user should not need any feature extraction or reduction techniques, as the recommended scheme was capable of learning the features without any human intervention, which in turn, saved time and effort.

7.3 Discussion

A deep learning model was suggested where the data were represented by a spectrogram and augmented before being transferred into the tensor domain. Two layers of the autoencoder were implemented by adjusting its parameters to gain optimal results. The SVM, ELM and SL were applied as the classifiers. The best local classifier was executed to select the highest accuracy per class. The proposed model was applied to different datasets to provide it with high fidelity, reliability and generalisation. Ten and six-finger movements were used for the recommended model compared to traditional pattern recognition system. The proposed model resulted in higher accuracies than the traditional pattern recognition. Moreover, pattern recognition consumed effort and time in extracting the best feature set that led to a higher accuracy. On the other hand, our model did not require any features or human interventions, as it was capable of learning features regardless of the introduced data set. The recommended model spent about 600 seconds to train the network and no more than 1.5 seconds to test the trained network. The planned model was applied to other widespread datasets where it achieved accepted accuracy values. As a future enhancement, the simulation time may be reduced by implementing different neural networks that may be able to learn features in a superior manner along with a shorter simulation time.

7.4 Summary

In this chapter, the data were augmented and represented in the tensor domain for deep learning implementation. The biosignals were represented by a spectrogram. For abundance purposes, the biosignals were augmented after spectrogram representation using the Gaussian algorithm. Afterwards, a tensor representation was applied to allow different dimensional representations of the biosignals before being introduced to the first layer of the autoencoder. The features were learnt in the two layer autoencoder, and passed on to the classification layer. Three-fold cross-validations were implemented. The extreme learning machine, support vector machine, and softmax layer were employed for classification. The last stage was classifier fusion, where the

best local classifier approach was applied. The implementation of the classifier fusion led to a higher accuracy value for either the training set or the testing set. Moreover, the proposed model was implemented for different datasets with different applications to ensure fidelity, reliability and generalisation. The six-finger movements for amputee participants were used in this study as well as five different datasets from the UCI machine learning repository. The classifier fusion was applied to the ten and six- finger movements only where the testing accuracy was higher than 90%. Furthermore, the five datasets were tested by the softmax layer classifier and achieved promising accuracy values. The chief objective of this thesis is to develop a deep learning model that is capable of adjusting to different biosignal applications, and learning the features without any human intervention or implementation of manual feature extraction methods. Moreover, it should be capable of achieving high accuracy values whilst keeping low simulation time to match real-time applications. Chapter 8 will summarise the thesis and will recommend future works.

CHAPTER 8

Summary and Conclusion

In this chapter, a summary of the research will be presented. In addition, recommendations for future research will be discussed.

8.1 Thesis Summary

The primary objective of this thesis is to introduce and develop a deep learning model that is capable of learning the features automatically. The model would not use any manual feature extraction approaches. Traditional feature extraction methodologies depend on user experiences and require significant time to reach the best set of features that match the application. Deep learning has numerous applications and is a major research area the field of machine learning. In this thesis, deep learning and feature learning models were presented and explained in detail.

In Chapter 4, an autoencoder was utilised as a feature learning algorithm to train the recommended model to learn useful features from input surface electromyography signals. The deep learning approach was implemented by using the autoencoder to learn the features. Wavelet packets, spectrogram, and wavelets were employed to represent the surface electromyography signals in the recommended model. Afterwards, the newly represented biosignals were introduced to the stacked autoencoder (2 stages) to learn features. The behaviour of the proposed algorithm was estimated by using different classifiers such as the extreme learning machine, support vector machine, and softMax layer. The rectified linear unit (ReLU) was estimated and used as an activation function for the extreme learning machine classifier, besides existing functions such as the sigmoid and radial basis functions. Both the confidence intervals and analysis of variance were estimated for the different classifiers. As a comparative study, signal representations were superimposed to compare the results with each representation individually. The classifier fusion layer was implemented to select the classifier

that led to the best accuracy values for both the testing and training sets to develop the results. The implemented dataset was collected from nine healthy subjects: seven males and two females. The age of the participants was between 24 years and 60 years. No subjects suffered from any muscle disorder, and they were normally limbed. The testing accuracy was more than 92% for the individual signal representation when implementing classifier fusion. It achieved 90% for the superimposed signal representations. by acquiring biosignal from two channels only. Furthermore, the extreme learning machine attained the narrowest interval, which was 2.53%. The widest one was 6.10% for the support vector machine. The analysis of variance was estimated with P value more than 0.05. The analysis of variance gives an indication of how the different classifiers vary from each other.

In Chapter 5, principal component analysis and independent component analysis were applied as a feature learning technique to train the model to learn features from input data. The deep learning approach was proposed by implementing either the PCA or ICA to learn features. The rationale behind using PCA and ICA for feature learning were the extended simulation time needed by the autoencoder deep learning model executed in Chapter 4. The autoencoder required a longer simulation time in the training phase. In contrast the PCA or ICA was expected to require a shorter time in the training stage. Therefore, both the PCA and ICA were implemented due to their short simulation time and their ability to deliver accurate values. In the proposed model, the raw data were read then represented by using different signal representations such as the spectrogram, wavelet and wavelet packets.

This represented data were then introduced to the PCA layer or ICA layer to learn features. The performance of this approach was evaluated by applying different classifiers including support vector machine, extreme learning machine and discriminant analysis classifiers. To improve the findings, the classifier fusion layer was used to select the most accurate result for both the training and testing sets. Moreover, the Manhattan index was estimated for the PCA learnt features to reduce the number of used features by selecting the highest 50 Manhattan index features.

The objective of assessing the Manhattan index is to use a lower number of features, which reduces simulation time while maintaining equivalent accuracy values. The dataset utilised in both models, either the PCA or ICA, was the exact dataset that was described in Chapter 4. For the PCA feature learning model, the testing accuracy was improved by adding the classifier fusion layer where the wavelet packets testing accuracy exceeded 90%. In contrast, the ICA feature learning model achieved testing accuracy higher than 86%. Furthermore, the Manhattan index was estimated for the PCA learnt features and the highest top 50 features were only included. The reduction of employed features reduced the simulation time spent in the training stage. Moreover, it did not impact on the testing accuracy values for the wavelet signal representation. The simulation time spent by either the PCA or ICA models was not comparable with the autoencoder. However, the testing accuracy for the PCA or ICA needed further improvements. Due to these requirements self-organising map for deep learning applications were implemented as discussed in Chapter 6.

In Chapter 6, the self-organising map was used as a feature learning technique to train the model on learning the features from the input data. The deep learning approach was proposed by deploying the SOM to learn the features. In the proposed model, the raw data were read then represented by using different signal representation as the spectrogram, wavelet and wavelet packets.

The represented data was the introduced to the self-organising map layer to generate features. The performance of the suggested scheme was evaluated by applying different classifiers (support vector machine, extreme learning machine, self-adaptive evolutionally extreme learning machine and discriminant analysis classification). The analysis of variance and confidence intervals for the different classifiers was calculated.

To further improve these findings, a classifier fusion layer was applied to select the most accurate result for both the training and testing set. The classifier fusion layer led to promising

training and testing accuracies. The dataset used in the self-organising map for the deep learning was the same dataset that was applied in Chapter 4. The testing accuracy value reached 94.44% for the wavelet packets along with the self-adaptive evolutionally extreme learning machine as a classifier. This result prior to executing the classifier fusion layer, achieved a testing accuracy of 96.6%. Furthermore, The ANOVA was estimated by a P value of 0.0613. The P value was not less than 0.05. This P value meant that there was no significant difference between the applied classifiers. By using the evolutionally extreme learning machine method it was nearly possible to achieve an ANOVA value close to 0.05. Moreover, the confidence intervals were calculated by a confidence score 60%.

As mentioned above, the narrowest interval was for the evolutionally extreme learning machine. The broadest range was for the support vector machine. These intervals provided an indication of how confident we were in the resulting testing accuracy.

Interestingly, if the testing accuracy was in the confidence interval range, this would suggest that the result was trustworthy by 60%. The SOM deep learning model was superior to either the autoencoder or PCA and ICA schemes in regards to the accuracy and simulation time. The simulation time spent by the SOM was much lower than that required by the autoencoder. In addition, the testing accuracy was higher than those achieved by the autoencoder, PCA or ICA. Eventually, a generalised model was presented in Chapter 7 with seven different datasets. The principal objective of the data-augmented deep learning model was to make the model adaptable to any input biosignals and, regardless of the application. This approach would achieve acceptable accuracies.

In Chapter 7, the deep learning concept was used by applying the autoencoder. The main contribution of this chapter was the approach of representing the biosignals. This provided the opportunity to examine different biosignals datasets, on the same proposed model, and achieve high accuracy results. The spectrogram of the biosignals was estimated, then the Gaussian representation for the spectrogram was used as a data augmentation phase. This augmented set

was represented in the tensor domain, which was introduced to the two-stage autoencoder to learn features. Ultimately, three different classifiers were used. These classifiers were the support vector machine, extreme learning machine, and Softmax layer. A classifier fusion layer was implemented to select the best local classifier. Furthermore, different datasets for different applications were introduced to the proposed model to claim generalisation.

It could be alleged that the suggested model was capable of learning features from biosignals for the different applications. The first dataset was that used in Chapter 4. This dataset was for nine healthy patients to classify ten-finger movements by using two channel electrodes. The second dataset was from amputee patients missing their left hand. The surface electromyography signals were recorded from eight channels at three levels of forces for nine amputees. The aim of collecting this data was to classify six different gestures to analyse and understand the biosignals controlling the upper limb prostheses.

The results of applying a typical feature extraction model for the ten and six-finger movements were compared with the suggested deep learning model for benchmarking purposes. Other datasets were imported from the UCI machine learning repository ".https://archive.ics.uci.edu/ml/datasets.html". The UCI website is archived and has been cited more than 1000 times by the machine learning community, including students or researchers. The performance of the proposed model was examined with five extra datasets that were archived at the UCI machine learning respiratory website. Those datasets included iris, breast cancer, seeds, sonar, mines vs rocks and Indian liver patients. The proposed deep learning model attained superior results to that achieved by the traditional pattern recognition scheme.

The results obtained by applying the suggested deep learning for the ten-finger movements was 90.25% for the testing accuracy. The testing accuracy only scored 87.11% for the traditional pattern recognition system. Furthermore, the testing accuracy was 91.85% for the deep learning implementation with the six finger gestures and achieved 89.64% for the same

datasets when applying the traditional pattern recognition scheme. The implementation of the iris data resulted in 100 % training accuracy and 98.5% testing accuracy. For the breast cancer tissue dataset, the training accuracy was 98.58% whereas the testing accuracy was 91.7%.

In regards to the sonar dataset, the accuracy of training was 85.69% and for testing was 74.4%. Moreover, applying the liver patients dataset led to 96.47% for the training accuracy and 85.1% for the testing. The seeds dataset, achieved a training accuracy of 94.57% whereas testing was 83.6%. The recommended model needed 600 seconds to train the network. It required no more than 1.5 seconds to test the trained network. The proposed model was applied to different datasets to provide it with fidelity, reliability and generalisation. Finally, the signal representation was critical step in presenting the biosignals to the proposed deep learning layer. When this was left out, the accuracy value was less than 50%.

8.2 Recommendations for future research

In this section, recommendations for future work will be discussed. These recommendations include improving the deep learning techniques, data fidelity and precision, developing the evaluation techniques, enhancing the classifiers performance, evolving the data acquisition approaches, targeting more classes, reducing the simulation time, and applying the outstanding models in real applications.

8.2.1 Improving deep learning techniques

Deep learning techniques should be investigated further as they can have promising results in regards to accuracy values. Deep learning provided the applications, which were presented in this thesis, with ease of learning the features. Currently, deep learning technology is invading different fields with various applications due to its advantages. Deep learning can be adapted by nearly all researchers and, this is due to the fact that deep learning learns the best features to match the application. Learning the features in a deep learning approach occurs without the need to apply any feature extraction or

reduction algorithms. Dispensing with all the feature extraction and reduction procedures saves simulation time and avoids entering the whirlpool of deciding which features should be extracted. The types of features are the time domain features, frequency domain features or mixed. The performance of any traditional pattern recognition system is significantly dependent on the extracted features that vary from one application to another. The answer to the question, which feature type should be included, can only be answered by the experts in the field after performing many trials to reach the best results. Furthermore, deep learning should be expanded to include many more fields as the same concept can be applied for different signals and various applications. The input may be mechanical, electrical or biomedical.It is highly recommended to apply any invented deep learning model on different applications with various datasets to guarantee trustworthiness, reliability and generalisation.

8.2.2 Data fidelity and precision

The fidelity and precision of the used datasets has a substantial impact on overall system performance. Accordingly, every effort must be made to secure a reliable data. More signal representation techniques should be investigated intensely. Further improvements need to be applied to signal acquisition devices with regards to its power cables and embedded filters. Regarding surface electromyography signal acquisition, the skin should be cleaned by the means of alcohol to remove any greasy layer. The electrodes should be positioned and fixed on the location of the targeted muscles while avoiding any shift in the electrode allocation or any movement of the participant during the signal collection. In addition, the electromyography signals are in the range of microvolts to millivolts, which makes amplification necessary. Further processing is needed by applying adequate filtering to remove the superimposed signals either from another muscle or unwanted noise. Simulation time should be considered while using filtering. Furthermore, the filtered electromyography signals should be well represented as this improved the results in this research. The researchers need to

explore innovative signal representations techniques to match the different signal applications. Moreover, it is highly recommended to develop high quality filters and power cables for the data acquisition devices to secure the lowest level of noise. Exploring a method which can identify the real biological data from the noise will be vital in the process of data mining and will assure higher quality system behavior.

8.2.3 Developing the evaluation techniques

In this type of research, classifier accuracy was the major gauge of pattern recognition or even the deep learning behavior. However, in real time applications the classifier is not the only scale by which the system performance is judged. This is one of the major reasons for the wide gap that exists between results from laboratory and real time applications. More aspects and parameters need to be addressed carefully during measuring the performance of any model. This needs to be done for either deep learning or traditional pattern recognition to reduce this gap between real time applications results and those conducted in the laboratory.

8.2.4 Enhancing classifier performance

Many classifiers exist today, however, their performance varies from one application to another. Therefore, different optimization techniques need to be addressed, investigated, explored and applied to the classifiers, besides classifier fusion, to lead to higher accuracy values and with shorter simulation time to match real application response time.

8.2.5 Evolving the data acquisition approaches

Most of this research was conducted on two channel datasets. However, collecting biosignals from more channels will improve the accuracy values as it will provide more details that are necessary. In real time applications, more related muscles are included than in laboratory research, where only few muscles are involved. This is considered as one of the foremost causes of the gap that exists between the outcomes from real

applications and the laboratory. Consequently, it is highly recommended and desirable to recruit and involve more related muscles in the study than shorten it to a limited number of muscles. This minimises the gap between the laboratory results and real time applications. However, dedicating more contiguous channels to collect electromyography signals from neighboring muscles may have a bad impact on the fidelity of collected signals. Therefore, this aspect should be considered in the design of the multi-channel devices.

8.2.6 Targeting more classes

For hand applications, it is highly recommended to classify as many classes as possible. This is to imitate real life applications and reduce the gap between their results and the laboratory results. As in the laboratory, only a limited number of gestures are classified, whereas in the real applications, many more movements are targeted. This causes a deterioration in system performance when it comes time to apply the simulated model to real time applications.

8.2.7 Reducing the simulation time

Simulation time should be taken into consideration to meet real applications' timing requirements. As for the hand movements, the real time applications accept up to 300 ms as a delay. It is recommended to consider simulation time into the design to avoid the collision with higher delays when it comes to apply the simulated model on the real time applications.

8.2.8 Applying outstanding models in real applications

It is recommended to execute the outstanding deep learning models in real life to measure how far or close its results are in comparison with the laboratory results. In this thesis, both the self-organizing map deep learning model and the augmented data variant achieved superior performance to the other schemes. This is in regards to the accuracy values, simulation time and generalization. The self-organizing map was very

distinguished in regards to accuracy and simulation time. The data augmented deep learning gained generalization and reliability as different datasets were applied to it. Subsequently, it is highly recommended to apply those two models in real applications, while taking into consideration the previous recommendations such as the multichannel acquired data, targeting more classes for the hand applications and collecting the surface electromyography signals by using high quality devices.

8.3 Conclusion

In this thesis, different deep learning and feature learning models were presented and then explored for their applications to hand movements. The first scheme was applying the autoencoder to learn features. Where two stages of the autoencoder were used. The signal representations were the impetus for improving the accuracy values. As applying different signal representations algorithms such as the spectrogram, wavelet and wavelet packets led the performance to a higher level. The classifier fusion was recruited to select the best local classifier. Moreover, the analysis of variance along with the confidence intervals were estimated. The testing accuracy value exceeded 92% to classify ten- finger movements by using a dataset collected from only two channels. However, the simulation time was not acceptable. Due to this extended simulation time, the next step was to try other feature learning methodologies. The principal component analysis and independent component analysis were performed to learn features from the same datasets that were used in the autoencoder. Both models achieved a lower simulation time, by more than ten times than that spent by the autoencoder. The testing accuracy exceeded 90% for the principal component analysis along with using a classifier fusion layer. The Manhattan index was estimated for the features learnt by the principal component analysis scheme and the top highest 50 features were included in the system while excluding the other features. The inclusion of these 50 features did not impact the accuracy values for the wavelet signal representation. The simulation time for the principal component analysis and independent component analysis were as a result shorter. Whereas the

accuracy values needed further improvement, which was the major drive to develop another deep learning scheme.

The self-organising map was implemented for the deep learning, and this achieved a superior result in regards to the accuracy and simulation time. The classifier fusion layer was utilised and achieved an accuracy value more than 96% with shorter simulation time. Moreover, the analysis of variance and the confidence intervals were estimated. Lastly, the self-organising map scheme was superior to the other implemented models. However, we needed to apply generalisation so that the model can be used with different biosignals and with various applications.

This idea prompted us to apply further signal representations and preparations than those applied in the previous schemes. The model that was presented in Chapter 7 had an advantage when compare to other schemes as it was applied on different datasets. Various datasets granted this approach increased the model fidelity and generalisation. As a result this model could be applied on different applications without consuming more time in extracting features. Moreover, the simulation on a trained network was not more than 1.5 seconds due to the applied signal representations approaches that were applied. The overall finding using this model approach achieved better accurate results with reduced amount of time and user input.

In summary, we put forward the idea that deep learning is more reliable than traditional feature extraction techniques. The deep learning can be a double-edged weapon, as it requires less user intervention. However, it may need longer simulation time than that consumed by training the network in learning the features. Therefore, researchers should work hard on finding new training techniques that require short simulation time to match real time applications along with achieving high accuracy values.

References

[1] I. K. Zola, "Toward the necessary universalizing of a disability policy," *The Milbank Quarterly,* pp. 401-428, 1989.

[2] G. L. Albrecht, K. D. Seelman, and M. Bury, *Handbook of disability studies*. Sage Publications, 2001.

[3] A. K. Mishra and R. Gupta, "Disability index: A measure of deprivation among disabled," *Economic and political weekly,* pp. 4026-4029, 2006.

[4] W. H. Organization, "World Health Day 2012: ageing and health: toolkit for event organizers," 2012.

[5] C. D. Mathers, A. D. Lopez, and C. J. Murray, "The burden of disease and mortality by condition: data, methods and results for 2001," *Global burden of disease and risk factors,* vol. 45, p. 88, 2006.

[6] M. Loeb, "International Census/Survey Data and the Short Set of Disability Questions Developed by the Washington Group on Disability Statistics," in *International Measurement of Disability*: Springer, 2016, pp. 255-304.

[7] N. Halfon, "Prevalence and impact of disabling chronic condition in childhood," *American Journal of Public Health,* vol. 88, pp. 610-17, 1998.

[8] C. M. Blackburn, J. M. Read, and N. J. Spencer, "Prevalence of childhood disability and the characteristics and circumstances of disabled children in the UK: secondary analysis of the Family Resources Survey," *BMC pediatrics,* vol. 10, no. 1, p. 21, 2010.

[9] I. O. Ertem and W. H. Organization, "Developmental difficulties in early childhood: prevention, early identification, assessment and intervention in low-and middle-income countries: a review," 2012.

[10] S. N. Zealand, "Disability counts 2001," *Wellington: Statistics New Zealand,* 2002.

[11] S. Savage and S. Bailey, "The impact of caring on caregivers' mental health: a review of the literature," *Australian health review,* vol. 27, no. 1, p. 111, 2004.

[12] K. Gerst-Emerson, R. Wong, A. Michaels-Obregon, and A. Palloni, "Cross-national differences in disability among elders: Transitions in disability in Mexico and the United States," *Journals of Gerontology Series B: Psychological Sciences and Social Sciences,* vol. 70, no. 5, pp. 759-768, 2015.

[13] U. Kriebernegg and R. Maierhofer, *The ages of life: living and aging in conflict?* transcript Verlag, 2014.

[14] TouchBionics, "The Big Picture: Bionic Hand," *IEEE Spectrum,* vol. 44, no. 10, pp. 22-22, 2007.

[15] P. Erik Scheme MSc and P. Kevin Englehart PhD, "Electromyogram pattern recognition for control of powered upper-limb prostheses: State of the art and challenges for clinical use," *Journal of rehabilitation research and development,* vol. 48, no. 6, p. 643, 2011.

[16] (2015). *TouchBionics*.

[17] J. Dunaj, W. J. Klimasara, Z. Pilat, and W. Rycerski, "Human-robot communication in rehabilitation devices," *Journal of Automation Mobile Robotics and Intelligent Systems,* vol. 9, 2015.

[18] A. D. Roche, H. Rehbaum, D. Farina, and O. C. Aszmann, "Prosthetic myoelectric control strategies: a clinical perspective," *Current Surgery Reports,* vol. 2, no. 3, p. 44, 2014.

[19] N. Uchida, A. Hiraiwa, N. Sonehara, and K. Shimohara, "EMG pattern recognition by neural networks for multi fingers control," in *Engineering in Medicine and Biology Society, 1992 14th Annual International Conference of the IEEE*, 1992, vol. 3, pp. 1016-1018: IEEE.

[20] R. N. Khushaba, S. Kodagoda, M. Takruri, and G. Dissanayake, "Toward improved control of prosthetic fingers using surface electromyogram (EMG) signals," *Expert Systems with Applications,* vol. 39, no. 12, pp. 10731-10738, 2012.

[21] M. A. Oskoei and H. Hu, "Support vector machine-based classification scheme for myoelectric control applied to upper limb," *IEEE transactions on biomedical engineering,* vol. 55, no. 8, pp. 1956-1965, 2008.

[22] A. H. Al-Timemy, G. Bugmann, J. Escudero, and N. Outram, "Classification of finger movements for the dexterous hand prosthesis control with surface electromyography," *IEEE Journal of Biomedical and Health Informatics,* vol. 17, no. 3, pp. 608-618, 2013.
[23] R. N. Khushaba, A. Al-Ani, and A. Al-Jumaily, "Orthogonal fuzzy neighborhood discriminant analysis for multifunction myoelectric hand control," *IEEE Transactions on Biomedical Engineering,* vol. 57, no. 6, pp. 1410-1419, 2010.
[24] N. Jiang, S. Dosen, K.-R. Muller, and D. Farina, "Myoelectric control of artificial limbs—is there a need to change focus?[In the spotlight]," *IEEE Signal Processing Magazine,* vol. 29, no. 5, pp. 152-150, 2012.
[25] F. V. Tenore, A. Ramos, A. Fahmy, S. Acharya, R. Etienne-Cummings, and N. V. Thakor, "Decoding of individuated finger movements using surface electromyography," *IEEE transactions on biomedical engineering,* vol. 56, no. 5, pp. 1427-1434, 2009.
[26] A. J. Young, L. J. Hargrove, and T. A. Kuiken, "Improving myoelectric pattern recognition robustness to electrode shift by changing interelectrode distance and electrode configuration," *IEEE Transactions on Biomedical Engineering,* vol. 59, no. 3, pp. 645-652, 2012.
[27] A. Wege and A. Zimmermann, "Electromyography sensor based control for a hand exoskeleton," in *Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on*, 2007, pp. 1470-1475: IEEE.
[28] T. L. Kivell, P. Lemelin, B. G. Richmond, and D. Schmitt, "The Evolution of the Primate Hand."
[29] M. L. Latash, *Synergy*. Oxford University Press, 2008.
[30] H.-M. Schmidt and U. Lanz, *Surgical anatomy of the hand*. Thieme, 2011.
[31] E. N. Marieb and K. Hoehn, *Human anatomy & physiology*. Pearson Education, 2007.
[32] M. Nordin and V. H. Frankel, *Basic biomechanics of the musculoskeletal system*. Lippincott Williams & Wilkins, 2001.
[33] P. Heo, G. M. Gu, S.-j. Lee, K. Rhee, and J. Kim, "Current hand exoskeleton technologies for rehabilitation and assistive engineering," *International Journal of Precision Engineering and Manufacturing,* vol. 13, no. 5, pp. 807-824, 2012.
[34] N. J. Engineer, R. Hazani, A. Mowlavi, M. W. Neumeister, W. A. Lee, and B. J. Wilhelmi, "Variations in the anatomy of the third common digital nerve and landmarks to avoid injury to the third common digital nerve with carpal tunnel release," *Eplasty,* vol. 8, 2008.
[35] B. Sahin and L. Seelig, "Arterial, neural and muscular variations in the upper limbs of a single cadaver," *Surgical and Radiologic Anatomy,* vol. 22, no. 5, pp. 305-308, 2001.
[36] H. P. von Schroeder and M. J. Botte, "Anatomy and functional significance of the long extensors to the fingers and thumb," *Clinical orthopaedics and related research,* vol. 383, pp. 74-83, 2001.
[37] J. E. Muscolino, *Kinesiology-E-Book: The Skeletal System and Muscle Function*. Elsevier Health Sciences, 2014.
[38] G. Windisch, "The fourth lumbrical muscle in the hand: a variation of insertion on the fifth finger," *Surgical and Radiologic Anatomy,* vol. 22, no. 3, pp. 213-215, 2000.
[39] T. J. Sullivan, S. R. Deiss, and G. Cauwenberghs, "A low-noise, non-contact EEG/ECG sensor," in *Biomedical Circuits and Systems Conference, 2007. BIOCAS 2007. IEEE*, 2007, pp. 154-157: IEEE.
[40] Y. M. Chi, P. Ng, E. Kang, J. Kang, J. Fang, and G. Cauwenberghs, "Wireless non-contact cardiac and neural monitoring," in *Wireless Health 2010*, 2010, pp. 15-23: ACM.
[41] C. Harland, T. Clark, and R. Prance, "Electric potential probes-new directions in the remote sensing of the human body," *Measurement Science and Technology,* vol. 13, no. 2, p. 163, 2001.
[42] C. Harland, T. Clark, N. Peters, M. J. Everitt, and P. Stiffell, "A compact electric potential sensor array for the acquisition and reconstruction of the 7-lead electrocardiogram without electrical charge contact with the skin," *Physiological measurement,* vol. 26, no. 6, p. 939, 2005.
[43] M. Oehler, V. Ling, K. Melhorn, and M. Schilling, "A multichannel portable ECG system with capacitive sensors," *Physiological measurement,* vol. 29, no. 7, p. 783, 2008.

[44] G. Kamen and E. Kinesiology, "Research methods in biomechanics," *Champaign, IL, Human Kinetics Publ,* 2004.
[45] C. De Luca, "Electromayography Encyclopedia of Medical Devices and Instrumentation," *John Wiley Publisher,* 2006.
[46] J. V. Basmajian and C. J. De Luca, *Muscles alive: their functions revealed by electromyography*. Williams & Wilkins, 1985.
[47] R. N. Khushaba, "Application of biosignal-driven intelligent systems for multifunction prosthesis control," 2010.
[48] P. Konrad, "The abc of emg," 2006.
[49] C. J. De Luca, "Surface electromyography: Detection and recording," *DelSys Incorporated,* vol. 10, p. 2011, 2002.
[50] R. M. Enoka, *Neuromechanical basis of kinesiology*. 1994.
[51] T. Moritani, D. Stegeman, and R. Merletti, "Basic physiology and biophysics of EMG signal generation," *Electromyography Physiology Engineering and Noninvasive Applications,* pp. 1-20, 2004.
[52] L. J. Hargrove, K. Englehart, and B. Hudgins, "A comparison of surface and intramuscular myoelectric signal classification," *IEEE transactions on biomedical engineering,* vol. 54, no. 5, pp. 847-853, 2007.
[53] J. R. Cram, *Cram's introduction to surface electromyography*. Jones & Bartlett Learning, 2010.
[54] P. Shenoy, K. J. Miller, J. G. Ojemann, and R. P. Rao, "Finger movement classification for an electrocorticographic BCI," in *Neural Engineering, 2007. CNE'07. 3rd International IEEE/EMBS Conference on*, 2007, pp. 192-195: IEEE.
[55] E. C. Leuthardt, K. J. Miller, G. Schalk, R. P. Rao, and J. G. Ojemann, "Electrocorticography-based brain computer interface-the Seattle experience," *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* vol. 14, no. 2, pp. 194-198, 2006.
[56] K. J. Miller *et al.*, "Electrocorticographic spectral changes with motor movement," *J Neurosci,* 2007.
[57] P. Shenoy, K. J. Miller, J. G. Ojemann, and R. P. Rao, "1 Generalized Features for Electrocorticographic BCIs," 2006.
[58] K. J. Miller, P. Shenoy, J. W. Miller, R. P. Rao, and J. G. Ojemann, "Real-time functional brain mapping using electrocorticography," *Neuroimage,* vol. 37, no. 2, pp. 504-507, 2007.
[59] T. M. Mitchell *et al.*, "Learning to decode cognitive states from brain images," *Machine learning,* vol. 57, no. 1, pp. 145-175, 2004.
[60] K.-R. Müller, M. Krauledat, G. Dornhege, G. Curio, and B. Blankertz, "Machine learning techniques for brain-computer interfaces," 2004.
[61] A. Zeghbib, F. Palis, and F. B. Ouezdou, "EMG-based Finger Movement Classification using transparent Fuzzy System," in *EUSFLAT Conf.*, 2005, pp. 816-821.
[62] H.-P. Huang, Y.-H. Liu, and C.-S. Wong, "Automatic EMG feature evaluation for controlling a prosthetic hand using supervised feature mining method: an intelligent approach," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, 2003, vol. 1, pp. 220-225: IEEE.
[63] R. Peck and J. L. Devore, *Statistics: The exploration & analysis of data*. Cengage Learning, 2011.
[64] H. J. Hermens *et al.*, "European recommendations for surface electromyography," *Roessingh research and development,* vol. 8, no. 2, pp. 13-54, 1999.
[65] L. Hargrove, K. Englehart, and B. Hudgins, "A training strategy to reduce classification degradation due to electrode displacements in pattern recognition based myoelectric control," *Biomedical signal processing and control,* vol. 3, no. 2, pp. 175-180, 2008.
[66] N. Sezgin, "A new hand finger movements' classification system based on bicoherence analysis of two-channel surface EMG signals," *Neural Computing and Applications,* pp. 1-11, 2017.
[67] M. Hinich and C. Clay, "The application of the discrete Fourier transform in the estimation of power spectra, coherence, and bispectra of geophysical data," *Reviews of geophysics,* vol. 6, no. 3, pp. 347-363, 1968.

[68] N. Sezgin, "Analysis of EMG signals in aggressive and normal activities by using higher-order spectra," *The Scientific World Journal,* vol. 2012, 2012.
[69] N. SEZGİN, "Epileptik EEG işaretlerin aşırı öğrenme makineleri ile sınıflandırılması," 2016.
[70] N. Sezgin, M. E. Tağluk, and M. Akın, "Using bispectral analysis in OSAS estimation," in *Signal Processing and Communications Applications Conference (SIU), 2010 IEEE 18th*, 2010, pp. 89-92: IEEE.
[71] N. Sezgin, "Nonlinear analysis of electrocardiography signals for atrial fibrillation," *The Scientific World Journal,* vol. 2013, 2013.
[72] M. Raghuveer and C. Nikias, "Bispectrum estimation: A parametric approach," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 33, no. 5, pp. 1213-1230, 1985.
[73] C. L. Nikias and J. M. Mendel, "Signal processing with higher-order spectra," *IEEE Signal processing magazine,* vol. 10, no. 3, pp. 10-37, 1993.
[74] K. Anam, R. N. Khushaba, and A. Al-Jumaily, "Two-channel surface electromyography for individual and combined finger movements," in *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, 2013, pp. 4961-4964: IEEE.
[75] K. Anam and A. Al-Jumaily, "Real-time Classification of Finger Movements using Two-channel Surface Electromyography," in *International Congress on Neurotechnology, Electronics and Informatics*, 2013: INSTICC.
[76] K. Anam and A. Al-Jumaily, "Swarm-wavelet based extreme learning machine for finger movement classification on transradial amputees," in *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, 2014, pp. 4192-4195: IEEE.
[77] K. Anam, A. Al-Jumaily, and Y. Maali, "Index Finger Motion Recognition using self-advise support vector machine," *International Journal on Smart Sensing and Intelligent Systems,* 2014.
[78] D. Tkach, H. Huang, and T. A. Kuiken, "Study of stability of time-domain features for electromyographic pattern recognition," *Journal of neuroengineering and rehabilitation,* vol. 7, no. 1, p. 21, 2010.
[79] K. Anam and A. Al-Jumaily, "Adaptive Wavelet Extreme Learning Machine (AW-ELM) for Index Finger Recognition Using Two-Channel Electromyography," in *International Conference on Neural Information Processing*, 2014, pp. 471-478: Springer.
[80] K. Anam and A. Al-Jumaily, "A novel extreme learning machine for dimensionality reduction on finger movement classification using sEMG," in *Neural Engineering (NER), 2015 7th International IEEE/EMBS Conference on*, 2015, pp. 824-827: IEEE.
[81] K. Anam and A. Al-Jumaily, "A robust myoelectric pattern recognition using online sequential extreme learning machine for finger movement classification," in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, 2015, pp. 7266-7269: IEEE.
[82] K. Anam and A. Al-Jumaily, "Evaluation of extreme learning machine for classification of individual and combined finger movements using electromyography on amputees and non-amputees," *Neural Networks,* vol. 85, pp. 51-68, 2017.
[83] R. N. Khushaba, S. Kodagoda, D. Liu, and G. Dissanayake, "Electromyogram (EMG) based fingers movement recognition using neighborhood preserving analysis with QR-decomposition," in *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2011 Seventh International Conference on*, 2011, pp. 1-105: IEEE.
[84] B. Hudgins, P. Parker, and R. N. Scott, "A new strategy for multifunction myoelectric control," *IEEE Transactions on Biomedical Engineering,* vol. 40, no. 1, pp. 82-94, 1993.
[85] B. Hjorth, "EEG analysis based on time domain properties," *Electroencephalography and clinical neurophysiology,* vol. 29, no. 3, pp. 306-310, 1970.
[86] R. N. Khushaba and S. Kodagoda, "Electromyogram (EMG) feature reduction using mutual components analysis for multifunction prosthetic fingers control," in *Control Automation Robotics & Vision (ICARCV), 2012 12th International Conference on*, 2012, pp. 1534-1539: IEEE.

[87] A. H. Al-Timemy, R. N. Khushaba, G. Bugmann, and J. Escudero, "Improving the performance against force variation of EMG controlled multifunctional upper-limb prostheses for transradial amputees," *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* vol. 24, no. 6, pp. 650-661, 2016.
[88] D. G. Childers, D. P. Skinner, and R. C. Kemerait, "The cepstrum: A guide to processing," *Proceedings of the IEEE,* vol. 65, no. 10, pp. 1428-1443, 1977.
[89] R. N. Khushaba, A. Al-Timemy, A. Al-Ani, and A. Al-Jumaily, "Myoelectric feature extraction using temporal-spatial descriptors for multifunction prosthetic hand control," in *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the*, 2016, pp. 1696-1699: IEEE.
[90] K. Englehart and B. Hudgins, "A robust, real-time control scheme for multifunction myoelectric control," *IEEE transactions on biomedical engineering,* vol. 50, no. 7, pp. 848-854, 2003.
[91] K. Englehart, B. Hudgin, and P. A. Parker, "A wavelet-based continuous classification scheme for multifunction myoelectric control," *IEEE Transactions on Biomedical Engineering,* vol. 48, no. 3, pp. 302-311, 2001.
[92] A. Phinyomark, 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.
[93] M. Zecca, S. Micera, M. C. Carrozza, and P. Dario, "Control of multifunctional prosthetic hands by processing the electromyographic signal," *Critical Reviews™ in Biomedical Engineering,* vol. 30, no. 4-6, 2002.
[94] M. Mouzé-Amady and F. Horwat, "Evaluation of Hjorth parameters in forearm surface EMG analysis during an occupational repetitive task," *Electroencephalography and Clinical Neurophysiology/Electromyography and Motor Control,* vol. 101, no. 2, pp. 181-183, 1996.
[95] K. Englehart, B. Hudgins, P. A. Parker, and M. Stevenson, "Classification of the myoelectric signal using time-frequency based representations," *Medical engineering & physics,* vol. 21, no. 6, pp. 431-438, 1999.
[96] A. M. Martínez and A. C. Kak, "Pca versus lda," *IEEE transactions on pattern analysis and machine intelligence,* vol. 23, no. 2, pp. 228-233, 2001.
[97] D. Cai, X. He, and J. Han, "SRDA: An efficient algorithm for large-scale discriminant analysis," *IEEE transactions on knowledge and data engineering,* vol. 20, no. 1, pp. 1-12, 2008.
[98] J. Ye, R. Janardan, Q. Li, and H. Park, "Feature reduction via generalized uncorrelated linear discriminant analysis," *IEEE Transactions on Knowledge and Data Engineering,* vol. 18, no. 10, pp. 1312-1322, 2006.
[99] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science,* vol. 313, no. 5786, pp. 504-507, 2006.
[100] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics,* vol. 2, no. 4, pp. 433-459, 2010.
[101] G. Tsenov, A. Zeghbib, F. Palis, N. Shoylev, and V. Mladenov, "Neural networks for online classification of hand and finger movements using surface EMG signals," in *Neural Network Applications in Electrical Engineering, 2006. NEUREL 2006. 8th Seminar on*, 2006, pp. 167-171: IEEE.
[102] A. Hiraiwa, K. Shimohara, and Y. Tokunaga, "EMG pattern analysis and classification by neural network," in *Systems, Man and Cybernetics, 1989. Conference Proceedings., IEEE International Conference on*, 1989, pp. 1113-1115: IEEE.
[103] C. Cipriani *et al.*, "Online myoelectric control of a dexterous hand prosthesis by transradial amputees," *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* vol. 19, no. 3, pp. 260-270, 2011.
[104] A. D. Chan and K. B. Englehart, "Continuous myoelectric control for powered prostheses using hidden Markov models," *IEEE Transactions on Biomedical Engineering,* vol. 52, no. 1, pp. 121-124, 2005.

[105] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* vol. 42, no. 2, pp. 513-529, 2012.
[106] Y. T. Y.-H. Pao, "Functional-link net computing: theory, system architecture, and functionalities," *IEEE Computer Society,* vol. 25, no. 5, pp. 76 - 79, 1992.
[107] Y.-H. Pao, G.-H. Park, and D. J. Sobajic, "Learning and generalization characteristics of the random vector functional-link net," *Neurocomputing,* vol. 6, no. 2, pp. 163-180, 1994.
[108] R. Fletcher, *Practical methods of optimization*. John Wiley & Sons, 2013.
[109] C. C. V. Vapnik, "Support-Vector Networks," *Machine Learning,* vol. 20, no. 3, pp. 273–297, 1995.
[110] I. A. Khalaf, A. M. Abualkishik, A. A. Aburomman, and M. B. IbneReaz, "Two Features Selection Algorithms based Onensemble of SVM Classifier For Intrusion Detection," *Australian Journal of Basic and Applied Sciences,* vol. 7, no. 7, pp. 480-485, 2013.
[111] W. S. Noble, "What is a support vector machine?," *NATURE BIOTECHNOLOGY* vol. 24, no. 12, pp. 1565-1567, 2006.
[112] K. C. Y. Singer, "On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines," *Machine Learning Research,* vol. 2, pp. 265-292, 2001.
[113] T. Hastie, Tibshirani, Robert, Friedman, Jerome, *The Elements of Statistical Learning*, Second ed. New York: Springer, 2008.
[114] N. Christianini, and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge , UK: Cambridge University Press, 2000.
[115] S. V. Yonas B. Dibike, Dimitri Solomatine and Michael B. Abbott, "MODEL INDUCTION WITH SUPPORT VECTOR MACHINES: INTRODUCTION AND APPLICATIONS," *JOURNAL OF COMPUTING IN CIVIL ENGINEERING* vol. 15, no. 3, pp. 208-216, 2001.
[116] S. T. a. D. Koller, "Support Vector Machine Active Learning with Applications to Text Classification," *Journal of Machine Learning Research,* pp. 45-66, 2001.
[117] B. S. a. T. H. Dwi Ratna "Bullet Image Classification using Support Vector Machine (SVM)," *Journal of Physics: Conference Series,* vol. 693, no. Conference1, 2016.
[118] C. M. Bishop, *Pattern Recognition and Machine Learning*. Cambridge, UK: Springer, 2006.
[119] J. W. Rui Zeng, Zhuhong Shao, Lotfi Senhadji and Huazhong Shu., "Quaternion softmax classifier," *Electronics Letters, IET,* vol. 50, no. 25, pp. 1929-1930, 2014.
[120] Z. Y. Zhehuan Zhao, Ling Luo, Hongfei Lin and Jian Wang, "Drug drug interaction extraction from biomedical literature using syntax convolutional neural network," *Bioinformatics,* vol. 32 no. 22, pp. 3444-3453, 2016.
[121] F. R. S. R. A. FISHER Sc.D., "THE USE OF MULTIPLE MEASUREMENTS IN TAXONOMIC PROBLEMS," *Annals of Human Genetics,* vol. 7, no. 2, pp. 179–188, 1936.
[122] G. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*. Wiley Interscience, 2004.
[123] D. Wetcher-Hendricks, *Analyzing Quantitative Data: An Introduction for Social Researchers*. John Wiley & Sons, 2011.
[124] S. C. C. a. Y. Rodin, *Statistical Methods of Discrimination and Classification Addvances in Theory and Applications*. USA: Pergamon, 1986, p. 143.
[125] Y. Q. Desheng Huang, Miao He and Baosen Zhou, "Comparison of linear discriminant analysis methods for the classification of cancer based on gene expression data," *Journal of Experimental & Clinical Cancer Research,* pp. 28-149, 2009.
[126] N. Hao, Dong, B. and Fan, J., "Sparsifying the Fisher linear discriminant by rotation " *Journal of the Royal Statistical Society. Series B: Statistical Methodology,* vol. 77, no. 4, pp. 827-851, 2015.
[127] T. Farrell and R. F. Weir, "Analysis window induced controller delay for multifunctional prostheses," 2008: Myoelectric Symposium.
[128] T. R. Farrell and R. F. Weir, "The optimal controller delay for myoelectric prostheses," *IEEE Transactions on neural systems and rehabilitation engineering,* vol. 15, no. 1, pp. 111-118, 2007.
[129] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*. Springer, 2013.

[130] M. L. Bermingham *et al.*, "Application of high-dimensional feature selection: evaluation for genomic prediction in man," *Scientific reports,* vol. 5, p. 10312, 2015.
[131] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on knowledge and data engineering,* vol. 17, no. 4, pp. 491-502, 2005.
[132] H. Liu and H. Motoda, *Feature selection for knowledge discovery and data mining*. Springer Science & Business Media, 2012.
[133] A. Y. Ng, "On feature selection: learning with exponentially many irrevelant features as training examples," Massachusetts Institute of Technology, 1998.
[134] J. Doak, "CSE-92-18-an evaluation of feature selection methodsand their application to computer security," 1992.
[135] B. Gilles and B. Paul, "Fundamentals of algorithms," ed: Prentice Hall Publication, 1996.
[136] M. Dash, K. Choi, P. Scheuermann, and H. Liu, "Feature selection for clustering-a filter solution," in *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, 2002, pp. 115-122: IEEE.
[137] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial intelligence,* vol. 97, no. 1-2, pp. 273-324, 1997.
[138] A. C. Yoshua Bengio, Pascal Vincent, "Representation Learning: A Review and New Perspectives," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,* vol. 35, no. 8, pp. 1798–1828, 2013.
[139] N. Srebro, J. Rennie, and T. S. Jaakkola, "Maximum-margin matrix factorization," in *Advances in neural information processing systems*, 2005, pp. 1329-1336.
[140] H. L. Adam Coates, Andrew Y. Ng, "An Analysis of Single-Layer Networks in Unsupervised Feature Learning," presented at the Artificial Intelligence and Statistics 2011.
[141] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on statistical learning in computer vision, ECCV*, 2004, vol. 1, no. 1-22, pp. 1-2: Prague.
[142] D. Jurafsky and J. H. Martin, *Speech and language processing*. Pearson London:, 2014.
[143] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach, "Supervised dictionary learning," in *Advances in neural information processing systems*, 2009, pp. 1033-1040.
[144] J. Turian, L. Ratinov, and Y. Bengio, "Word representations: a simple and general method for semi-supervised learning," in *Proceedings of the 48th annual meeting of the association for computational linguistics*, 2010, pp. 384-394: Association for Computational Linguistics.
[145] F. Schwenker, H. A. Kestler, and G. Palm, "Three learning phases for radial-basis-function networks," *Neural networks,* vol. 14, no. 4-5, pp. 439-458, 2001.
[146] A. Coates and A. Y. Ng, "Learning feature representations with k-means," in *Neural networks: Tricks of the trade*: Springer, 2012, pp. 561-580.
[147] D. Lin and X. Wu, "Phrase clustering for discriminative learning," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, 2009, pp. 1030-1038: Association for Computational Linguistics.
[148] L. K. Saul and S. T. Roweis, "An introduction to locally linear embedding," *unpublished. Available at: http://www. cs. toronto. edu/~ roweis/lle/publications. html,* 2000.
[149] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural networks,* vol. 13, no. 4-5, pp. 411-430, 2000.
[150] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Advances in neural information processing systems*, 2007, pp. 801-808.
[151] M. Aharon, M. Elad, and A. Bruckstein, "$ rm k $-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on signal processing,* vol. 54, no. 11, pp. 4311-4322, 2006.
[152] L. Y. Deng, D, "Deep Learning: Methods and Applications," *Foundations and Trends in Signal Processing,* vol. 7, pp. 3-4, 2014.
[153] Y. Bengio, "Learning Deep Architectures for AI," *Foundations and Trends in Machine Learning,* vol. 2, no. 1, pp. 1-127, 2009.

[154] Y. C. Bengio, A.; Vincent, P., "Representation Learning: A Review and New Perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 35, no. 8, pp. 1798–1828, 2013.
[155] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks,* vol. 61, pp. 85-117, 2015.
[156] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature,* vol. 521, no. 7553, pp. 436-444, 2015.
[157] I. Arel, D. C. Rose, and T. P. Karnowski, "Deep machine learning-a new frontier in artificial intelligence research [research frontier]," *IEEE Computational Intelligence Magazine,* vol. 5, no. 4, pp. 13-18, 2010.
[158] H. A. Song and S.-Y. Lee, "Hierarchical representation using NMF," in *International Conference on Neural Information Processing*, 2013, pp. 466-473: Springer.
[159] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature,* vol. 381, no. 6583, p. 607, 1996.
[160] A. G. e. Ivakhnenko and V. G. Lapa, "Cybernetic predicting devices," DTIC Document1966.
[161] A. G. Ivakhnenko, "Polynomial theory of complex systems," *IEEE transactions on Systems, Man, and Cybernetics,* vol. 1, no. 4, pp. 364-378, 1971.
[162] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and cooperation in neural nets*: Springer, 1982, pp. 267-285.
[163] Y. LeCun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural computation,* vol. 1, no. 4, pp. 541-551, 1989.
[164] A. De Carvalho, M. C. Fairhurst, and D. L. Bisset, "An integrated Boolean neural network for pattern classification," *Pattern recognition letters,* vol. 15, no. 8, pp. 807-813, 1994.
[165] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature,* vol. 323, no. 6088, p. 533, 1986.
[166] S. Hochreiter, "Untersuchungen zu dynamischen neuronalen Netzen," diploma thesis, institut für informatik, lehrstuhl prof. brauer, technische universität münchen, 1991.
[167] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," ed: A field guide to dynamical recurrent neural networks. IEEE Press, 2001.
[168] N. Morgan, H. Bourlard, S. Renals, M. Cohen, and H. Franco, "Hybrid neural network/hidden Markov model systems for continuous speech recognition," *International Journal of Pattern Recognition and Artificial Intelligence,* vol. 7, no. 04, pp. 899-916, 1993.
[169] T. Robinson, "A real-time recurrent error propagation network word recognition system," in *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, 1992, vol. 1, pp. 617-620: IEEE.
[170] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE transactions on acoustics, speech, and signal processing,* vol. 37, no. 3, pp. 328-339, 1989.
[171] J. M. Baker *et al.*, "Developments and directions in speech recognition and understanding, Part 1 [DSP Education]," *IEEE Signal Processing Magazine,* vol. 26, no. 3, 2009.
[172] Y. Bengio, "Artificial neural networks and their application to sequence recognition," 1991.
[173] L. Deng, K. Hassanein, and M. Elmasry, "Analysis of the correlation structure for a neural predictive model with application to speech recognition," *Neural Networks,* vol. 7, no. 2, pp. 331-339, 1994.
[174] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation,* vol. 9, no. 8, pp. 1735-1780, 1997.
[175] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369-376: ACM.
[176] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Advances in Neural Information Processing Systems*, 2013, pp. 2553-2561.

[177] F. A. Gers and E. Schmidhuber, "LSTM recurrent networks learn simple context-free and context-sensitive languages," *IEEE Transactions on Neural Networks,* vol. 12, no. 6, pp. 1333-1340, 2001.
[178] H. J. Kelley, "Gradient theory of optimal flight paths," *Ars Journal,* vol. 30, no. 10, pp. 947-954, 1960.
[179] E. Mizutani, S. E. Dreyfus, and K. Nishio, "On derivation of MLP backpropagation from the Kelley-Bryson optimal-control gradient formula and its application," in *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, 2000, vol. 2, pp. 167-172: IEEE.
[180] E. A. Wan, "Time series prediction by using a connectionist network with internal delay lines," in *SANTA FE INSTITUTE STUDIES IN THE SCIENCES OF COMPLEXITY-PROCEEDINGS VOLUME-*, 1993, vol. 15, pp. 195-195: Addison-Wesley publishing co.
[181] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine,* vol. 29, no. 6, pp. 82-97, 2012.
[182] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013, pp. 8609-8613: IEEE.
[183] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016.
[184] N. A. a. T. S. H. J. Weng, "Learning recognition and segmentation of 3-D objects from 2-D images," in *Proc. 4th International Conf. Computer Vision*, Berlin, Germany, 1993, pp. 121-128.
[185] S. Linnainmaa, "The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors," *Master's Thesis (in Finnish), Univ. Helsinki,* pp. 6-7, 1970.
[186] C. Goller and A. Kuchler, "Learning task-dependent distributed representations by backpropagation through structure," in *Neural Networks, 1996., IEEE International Conference on*, 1996, vol. 1, pp. 347-352: IEEE.
[187] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng, "Parsing natural scenes and natural language with recursive neural networks," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 129-136.
[188] R. Socher *et al.*, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, 2013, vol. 1631, p. 1642.
[189] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation,* vol. 14, no. 8, pp. 1771-1800, 2002.
[190] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of machine learning research,* vol. 3, no. Feb, pp. 1137-1155, 2003.
[191] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2012, pp. 3642-3649: IEEE.
[192] G. E. Dahl, N. Jaitly, and R. Salakhutdinov, "Multi-task neural networks for QSAR predictions," *arXiv preprint arXiv:1406.1231,* 2014.
[193] Y. Tkachenko, "Autonomous CRM control via CLV approximation with deep reinforcement learning in discrete and continuous action space," *arXiv preprint arXiv:1504.01840,* 2015.
[194] A. Van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," in *Advances in neural information processing systems*, 2013, pp. 2643-2651.
[195] D. Chicco, P. Sadowski, and P. Baldi, "Deep autoencoder neural networks for gene ontology annotation predictions," in *Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, 2014, pp. 533-540: ACM.
[196] D. O. Hebb, *The organization of behavior: A neuropsychological approach*. John Wiley & Sons, 1949.
[197] W. S. McCulloch, "A logical calculus of ideas imminent in nervous activity," *Biol Math. Biophys,* 1943.

[198] S. Kleene, "Representation of Events in Nerve Nets and Finite Automata," 1956.
[199] B. Farley and W. Clark, "Simulation of self-organizing systems by digital computer," *Transactions of the IRE Professional Group on Information Theory,* vol. 4, no. 4, pp. 76-84, 1954.
[200] N. Rochester, J. Holland, L. Haibt, and W. Duda, "Tests on a cell assembly theory of the action of the brain, using a large digital computer," *IRE Transactions on information Theory,* vol. 2, no. 3, pp. 80-93, 1956.
[201] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological review,* vol. 65, no. 6, p. 386, 1958.
[202] P. Werbos, "Beyond regression: New tools for prediction and analysis in the behavioral sciences," 1974.
[203] M. Minsky and S. Papert, "Perceptrons: an introduction to computational geometry (expanded edition)," ed: MIT Press, Cambridge, Ma, 1988.
[204] A. Zell *et al.*, "SNNS (stuttgart neural network simulator)," in *Neural Network Simulation Environments*: Springer, 1994, pp. 165-186.
[205] P. K. Simpson, "Foundations of neural networks," USDOE Pittsburgh Energy Technology Center, PA (United States); Oregon State Univ., Corvallis, OR (United States). Dept. of Computer Science; Naval Research Lab., Washington, DC (United States); Electric Power Research Inst., Palo Alto, CA (United States); Bureau of Mines, Washington, DC (United States)1994.
[206] S. Haykin and N. Network, "A comprehensive foundation," *Neural networks,* vol. 2, no. 2004, p. 41, 2004.
[207] V. K. Ojha, A. Abraham, and V. Snášel, "Metaheuristic design of feedforward neural networks: A review of two decades of research," *Engineering Applications of Artificial Intelligence,* vol. 60, pp. 97-116, 2017.
[208] S. Dominic, R. Das, D. Whitley, and C. Anderson, "Genetic reinforcement learning for neural networks," in *Neural Networks, 1991., IJCNN-91-Seattle International Joint Conference on*, 1991, vol. 2, pp. 71-76: IEEE.
[209] J. Hoskins and D. Himmelblau, "Process control via artificial neural networks and reinforcement learning," *Computers & chemical engineering,* vol. 16, no. 4, pp. 241-251, 1992.
[210] D. P. Bertsekas and J. N. Tsitsiklis, "Neuro-dynamic programming: an overview," in *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*, 1995, vol. 1, pp. 560-564: IEEE.
[211] N. Secomandi, "Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands," *Computers & Operations Research,* vol. 27, no. 11-12, pp. 1201-1225, 2000.
[212] D. De Rigo, A. E. Rizzoli, R. Soncini-Sessa, E. Weber, and P. Zenesi, "Neuro-dynamic programming for the efficient management of reservoir networks," 2001.
[213] M. Damas, M. Salmerón, A. Diaz, J. Ortega, A. Prieto, and G. Olivares, "Genetic algorithms and neuro-dynamic programming: application to water supply networks," in *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, 2000, vol. 1, pp. 7-14: IEEE.
[214] G. Deng and M. C. Ferris, "Neuro-dynamic programming for fractionated radiotherapy planning," in *Optimization in medicine*: Springer, 2008, pp. 47-70.
[215] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 160-167: ACM.
[216] A. A. A.-J. Marwa Farouk Ibrahim, "Auto-Encoder based Deep Learning for Surface Electromyography Signal Processing," *Advances in Science, Technology and Engineering Systems Journal (ASTESJ),* vol. 1, no. 3, pp. 94-102, 2018.
[217] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114,* 2013.
[218] A. Larsen and S. Sønderby, "Generating faces with Torch, Nov 2015," ed.
[219] C.-Y. Liou, J.-C. Huang, and W.-C. Yang, "Modeling word perception using the Elman network," *Neurocomputing,* vol. 71, no. 16-18, pp. 3150-3157, 2008.

[220] C.-Y. Liou, W.-C. Cheng, J.-W. Liou, and D.-R. Liou, "Autoencoder for words," *Neurocomputing,* vol. 139, pp. 84-96, 2014.
[221] Y. Zeinali and B. A. Story, "Competitive probabilistic neural network," *Integrated Computer-Aided Engineering,* vol. 24, no. 2, pp. 105-118, 2017.
[222] D. F. Specht, "Probabilistic neural networks," *Neural networks,* vol. 3, no. 1, pp. 109-118, 1990.
[223] V. Cheung and K. Cannons, "An introduction to probabilistic neural networks," *Signal & Data Compression Laboratory, Electrical & Computer Engineering, University of Manitoba, Winnipeg, Manitoba, Canada, www. psi. toronto. edu/~ vincent/research/presentations/P NN. pdf,* 2003.
[224] M. Matsugu, K. Mori, Y. Mitari, and Y. Kaneda, "Subject independent facial expression recognition with robust face detection using a convolutional neural network," *Neural Networks,* vol. 16, no. 5-6, pp. 555-559, 2003.
[225] Y. LeCun, "LeNet-5, convolutional neural networks," *URL: http://yann. lecun. com/exdb/lenet,* p. 20, 2015.
[226] H. H. Aghdam and E. J. Heravi, *Guide to Convolutional Neural Networks: A Practical Application to Traffic-Sign Detection and Classification*. Springer, 2017.
[227] D. Lowe, "Multi-variable functional interpolation and adaptive networks," *Complex Systems,* vol. 2, pp. 321-355, 1988.
[228] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics,* vol. 43, no. 1, pp. 59–69, 1982.
[229] T. Kohonen, "The self-organizing map," *Neurocomputing,* vol. 21, no. 1-3, pp. 1-6, 1998.
[230] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE,* vol. 78, no. 9, pp. 1464-1480, 1990.
[231] M. F. I. Ibrahim and A. A. Al-Jumaily, "Self-Organizing Map based Feature Learning in Bio-Signal Processing," 2017.
[232] I. D. Ervin Sejdic, Jin Jiang "Time-frequency feature representation using energy concentration: An overview of recent advances," *Digital Signal Processing,* vol. 19, no. 1, pp. 153-183, 2009.
[233] R. R. Mergu and S. K. Dixit, "Multi-resolution speech spectrogram," *International Journal of Computer Applications,* vol. 15, no. 4, pp. 28-32, 2011.
[234] S. A. Fulop and K. Fitz, "Algorithms for computing the time-corrected instantaneous frequency (reassigned) spectrogram, with applications," *The Journal of the Acoustical Society of America,* vol. 119, no. 1, pp. 360-371, 2006.
[235] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *Information Theory,* vol. 36, no. 5, pp. 961 - 1005, 1990.
[236] M. Farye, "WAVELET TRANSFORMS AND THEIR APPLICATIONS TO TURBULENCE," *Fluid Mechanics,* vol. 24, pp. 395–457, 1992.
[237] H. W. a. K.-M. Lau, "Wavelets, Period Doubling, and Time–Frequency Localization with Application to Organization of Convection over the Tropical Western Pacific," *Journal of the Atmospheric Sciences,* vol. 51, no. 17, pp. 2523–2541., 1994.
[238] D. G. a. S. G. H. Philander, "Secular Changes of Annual and Interannual Variability in the Tropics during the Past Century," *Journal of Climate,* vol. 8, pp. 864-876, 1995.
[239] B. W. a. Y. Wang, "Temporal Structure of the Southern Oscillation as Revealed by Waveform and Wavelet Analysis," *Journal of Climate,* vol. 9, pp. 1586-1598, 1996.
[240] N. Gamage, and W. Blumen, "Comparative analysis of lowlevel cold fronts: Wavelet, Fourier, and empirical orthogonal function decompositions," *Monthly Weather Review,* vol. 121, pp. 2867–2878, 1993.
[241] P. F. Sallie Baliunas, Dmitry Sokoloff and Willie Soon, "Time scales and trends in the central England temperature data (1659–1990)," *BrAVELET ANALYSIS OF CENTRAL ENGLAND TEMPERATURE* vol. 24, pp. 1351-1354, 1997.
[242] A. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, "Wavelet transforms that map integers to integers," *Applied and computational harmonic analysis,* vol. 5, no. 3, pp. 332-369, 1998.

[243] J. Lin and L. Qu, "Feature extraction based on Morlet wavelet and its application for mechanical fault diagnosis," *Journal of sound and vibration,* vol. 234, no. 1, pp. 135-148, 2000.
[244] J. Lin and M. Zuo, "Gearbox fault diagnosis using adaptive wavelet filter," *Mechanical systems and signal processing,* vol. 17, no. 6, pp. 1259-1269, 2003.
[245] M. V. Wickerhauser, "INRIA lectures on wavelet packet algorithms," 1991.
[246] R. P. a. G. Wilson, "Application of 'matched' wavelets to identification of metallic transients," in *Proceedings of the IEEE-SP International Symposium Time-Frequency and Time-Scale Analysis*, Victoria, BC, Canada, Canada, 1992: IEEE.
[247] E. S. a. M. Fabio, "The use of the discrete wavelet transform for acoustic emission signal processing," in *Proceedings of the IEEE–SP International Symposium*, Victoria, British Columbia, Canada, 1992: IEEE.
[248] T. P. T. Brotherton, R. Barton, A. Krieger, and L. Marple, "Applications of time frequency and time scale analysis to underwater acoustic transients," in *Proceedings of the IEEE–SP International Symposium*, Victoria, British Columbia, Canada, 1992: IEEE.
[249] M. Y. G. a. D. K. Khanduj, "Time Domain Signal Analysis Using Wavelet Packet Decomposition Approach " *Int. J. Communications, Network and System Sciences,* vol. 3, pp. 321-329, 2010.
[250] J.-G. Han, W.-X. Ren, and Z.-S. Sun, "Wavelet packet based damage identification of beam structures," *International Journal of Solids and Structures,* vol. 42, no. 26, pp. 6610-6627, 2005.
[251] J.-D. Wu and C.-H. Liu, "An expert system for fault diagnosis in internal combustion engines using wavelet packet transform and neural network," *Expert systems with applications,* vol. 36, no. 3, pp. 4278-4286, 2009.
[252] D. A. Van Dyk and X.-L. Meng, "The art of data augmentation," *Journal of Computational and Graphical Statistics,* vol. 10, no. 1, pp. 1-50, 2001.
[253] S. C. Wong, A. Gatt, V. Stamatescu, and M. D. McDonnell, "Understanding data augmentation for classification: when to warp?," in *Digital Image Computing: Techniques and Applications (DICTA), 2016 International Conference on*, 2016, pp. 1-6: IEEE.
[254] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097-1105.
[255] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556,* 2014.
[256] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1-9.
[257] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.
[258] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *ICDAR*, 2003, vol. 3, pp. 958-962.
[259] D. C. Cireşan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep, big, simple neural nets for handwritten digit recognition," *Neural computation,* vol. 22, no. 12, pp. 3207-3220, 2010.
[260] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of artificial intelligence research,* vol. 16, pp. 321-357, 2002.
[261] H. S. Baird, "Document image defect models," in *Structured Document Image Analysis*: Springer, 1992, pp. 546-556.
[262] L. S. Yaeger, R. F. Lyon, and B. J. Webb, "Effective training of a neural network character classifier for word recognition," in *Advances in Neural Information Processing Systems*, 1997, pp. 807-816.
[263] T. Poggio and T. Vetter, "Recognition and structure from one 2D model view: Observations on prototypes, object classes and symmetries," MASSACHUSETTS INST OF TECH CAMBRIDGE ARTIFICIAL INTELLIGENCE LAB1992.

[264] B. Schölkopf, P. Simard, A. J. Smola, and V. Vapnik, "Prior knowledge in support vector kernels," in *Advances in neural information processing systems*, 1998, pp. 640-646.
[265] J.-S. Su, B.-F. Zhang, and X. Xu, "Advances in machine learning based text categorization," *Ruan Jian Xue Bao(Journal of Software),* vol. 17, no. 9, pp. 1848-1859, 2006.
[266] X. Wang, L. Guo, and J. Fang, "Research on ontology-driven text virtual sample constructing," *Computer Science,* vol. 35, no. 3, pp. 142-145, 2008.
[267] J. Wen, S. Luo, J. Zhao, and H. Huang, "A small sample face recognition statistical learning method based on virtual samples," *Computer Research and Development,* vol. 39, no. 7, pp. 814-818, 2002.
[268] R. Xu, L. He, L. Zhang, Z. Tang, and S. Tu, "Research on virtual sample based identification of noise sources in ribbed cylindrical double-shells," *Journal of Vibration and Shock,* vol. 27, no. 5, pp. 32-35, 2008.
[269] L. Zhang and G.-H. Chen, "Method for constructing training data set in intrusion detection system," *Jisuanji Gongcheng yu Yingyong(Computer Engineering and Applications),* vol. 42, no. 28, pp. 145-146, 2006.
[270] J. Yang, X. Yu, Z.-Q. Xie, and J.-P. Zhang, "A novel virtual sample generation method based on Gaussian distribution," *Knowledge-Based Systems,* vol. 24, no. 6, pp. 740-748, 2011.
[271] T. K. Ho, J. J. Hull, and S. N. Srihari, "Decision combination in multiple classifier systems," *IEEE transactions on pattern analysis and machine intelligence,* vol. 16, no. 1, pp. 66-75, 1994.
[272] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," *Neural computation,* vol. 6, no. 2, pp. 181-214, 1994.
[273] K. Woods, W. P. Kegelmeyer, and K. Bowyer, "Combination of multiple classifiers using local accuracy estimates," *IEEE transactions on pattern analysis and machine intelligence,* vol. 19, no. 4, pp. 405-410, 1997.
[274] P. D. Gader, M. A. Mohamed, and J. M. Keller, "Fusion of handwritten word classifiers," *Pattern Recognition Letters,* vol. 17, no. 6, pp. 577-584, 1996.
[275] H. Tahani and J. Keller, "Information Fusion in Computer Vision Using Fuzzy Integral Operator," *IEEE Trans on Systems, Man and Cybernetics,* vol. 20, 1994.
[276] G. E. Hinton, "Products of experts," 1999.
[277] L. Xu, A. Krzyzak, and C. Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE transactions on systems, man, and cybernetics,* vol. 22, no. 3, pp. 418-435, 1992.
[278] S. Theodoridis and K. Koutroumbas, "Pattern Recognition (San Diego, CA: Academic)," 1999.
[279] G. J. Klir and T. A. Folger, *Fuzzy sets, uncertainty, and information*. Prentice Hall Englewood Cliffs, 1988.
[280] G. E. H. a. R. R. Salakhutdinov. (2006) Reducing the Dimensionality of Data with Neural Networks. *SCIENCE*.
[281] A. H. Noriyoshi Uchida, Noboru Sonehara, and Katsunori Shimohara, "EMG pattern recognition by neural networks for multi fingers control," presented at the Engineering in Medicine and Biology Society, 1992 14th Annual International Conference of the IEEE, 1992.
[282] E. M. a. L. M. Alex Andrews, "Optimal Electrode Configurations for Finger Movement Classification using EMG," presented at the 31st Annual International Conference of the IEEE EMBS, 2009.
[283] S. K. Rami N. Khushaba, Dikai Liu, Gamini Dissanayake, "Electromyogram (EMG) Based Fingers Movement Recognition Using Neighborhood Preserving Analysis with QR-Decomposition," presented at the 2011 Seventh International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2011.
[284] C. A. a. C. C. Gunter R. Kanitz, "Decoding of Individuated Finger Movements Using Surface EMG and Input Optimization Applying a Genetic Algorithm," presented at the 33rd Annual International Conference of the IEEE EMBS, Boston, Massachusetts USA, 2011.
[285] G. B. Ali H. Al-Timemy, Javier Escudero and Nicholas Outram, "Classification of Finger Movements for the Dexterous Hand Prosthesis Control With Surface Electromyography,"

IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS, vol. 17, no. 03, pp. 608-618, 2013.

[286] W. C. Mochammad Ariyanto, Khusnul A. Mustaqim, Mohamad Irfan, Jonny A. Pakpahan, Joga D. Setiawan and Andri R. Winoto, "Finger Movement Pattern Recognition Method Using Artificial Neural Network Based on Electromyography (EMG) Sensor," presented at the 2015 International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT), Bandung, Indonesia, 2015.

[287] C. W. Daniele Rav`ı, Fani Deligianni, Melissa Berthelot, Javier Andreu-Perez, Benny Lo and Guang-Zhong Yang, "Deep Learning for Health Informatics," *IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS,* vol. 21, no. 1, pp. 4-21, 2016.

[288] T. K. Julius Hannink, Cristian F. Pasluosta, Karl-G ¨unter Gaßmann, Jochen Klucken, and Bjoern M. Eskofier, "Sensor-Based Gait Parameter Extraction With Deep Convolutional Neural Networks," *IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS,* vol. 21, no. 1, pp. 85-93, 2016.

[289] T. T. Phuoc Nguyen, Nilmini Wickramasinghe, and Svetha Venkatesh, "Deepr: A Convolutional Net for Medical Records," *IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS,* vol. 21, no. 1, pp. 22-30, 2016.

[290] C. W. Daniele Rav`ı, Benny Lo, and Guang-Zhong Yang, "A Deep Learning Approach to on-Node Sensor Data Analytics for Mobile or Wearable Devices," *IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS,* vol. 21, no. 1, pp. 56-64, 2016.

[291] T.-S. Pierre Comon, Parc Soph & Antipolis, "Independent component analysis, A new concept?," *Signal Processing,* vol. 36, no. 3, pp. 287–314, 1994.

[292] A. W. Bronkhorst, "The Cocktail Party Phenomenon: A Review on Speech Intelligibility in Multiple-Talker Conditions," *Acta Acustica united with Acustica,* vol. 86, pp. 117-128, 2000.

[293] C. N. Wood N, "The cocktail party phenomenon revisited: how frequent are attention shifts to one's name in an irrelevant auditory channel?," *J Exp Psychol Learn Mem Cogn,* vol. 21, no. 1, pp. 255-60, 1995.

[294] C. N. Conway AR, Bunting MF "The cocktail party phenomenon revisited: the importance of working memory capacity," *Psychon Bull Rev,* vol. 8, no. 2, pp. 331–335, 2001.

[295] G. E. Hinton, "Deep belief networks," *Scholarpedia,* vol. 4, no. 5, p. 5947, 2009.

[296] Z. Q.-Y. Huang G-B, Siew C-K, "Extreme learning machine: a new learning scheme of feedforward neural networks," *In: Proceedings of international joint conference on neural networks, Budapest, Hungary,,* pp. 985–990, 2004.

[297] Z. Q.-Y. Huang G-B, Siew C-K, "Extreme learning machine: theory and applications.," *Neurocomputing,* vol. 70, no. 1-3, pp. 489–501, 2006.

[298] J. Tang, C. Deng, and G. B. Huang, "Extreme Learning Machine for Multilayer Perceptron," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 27, no. 4, pp. 809-821, 2016.

[299] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, and A. Lendasse, "OP-ELM: Optimally Pruned Extreme Learning Machine," *IEEE Transactions on Neural Networks,* vol. 21, no. 1, pp. 158-162, 2010.

[300] J. Cao, Z. Lin, and G.-B. Huang, "Self-Adaptive Evolutionary Extreme Learning Machine," *Springer Science+Business Media, LLC. 2012,* 18 July 2012 2012.

[301] L. D. S. Pacifico and T. B. Ludermir, "Evolutionary extreme learning machine based on particle swarm optimization and clustering strategies," in *Neural Networks (IJCNN), The 2013 International Joint Conference on*, 2013, pp. 1-6.

[302] N. Zhang, Y. Qu, and A. Deng, "Evolutionary Extreme Learning Machine Based Weighted Nearest-Neighbor Equality Classification," in *2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics*, 2015, vol. 2, pp. 274-279.

[303] K. F. Man, K. S. Tang, and S. Kwong, "Genetic algorithms: concepts and applications [in engineering design]," *IEEE Transactions on Industrial Electronics,* vol. 43, no. 5, pp. 519-534, 1996.

[304] J. Shi, Y. Yang, P. Wang, Y. Liu, and S. Han, "Genetic algorithm-piecewise support vector machine model for short term wind power prediction," in *Intelligent Control and Automation (WCICA), 2010 8th World Congress on*, 2010, pp. 2254-2258: IEEE.

[305] S. K. Rami N. Khushaba, Dikai Liu, Gamini Dissanayake, "Electromyogram (EMG) Based Fingers Movement Recognition Using Neighborhood Preserving Analysis with QR-Decomposition," presented at the 2011 Seventh International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Adelaide, SA, Australia, 2011.
[306] R. N. K. a. A. A.-J. Khairul Anam, "Two-Channel Surface Electromyography for Individual and Combined Finger Movements," presented at the 35th Annual International Conference of the IEEE EMBS, Osaka, Japan, 2013.
[307] K. A. a. A. Al-Jumaily, "Swarm-based Extreme Learning Machine for Finger Movement Recognition," presented at the 2014 Middle East Conference on Biomedical Engineering (MECBME), Doha, Qatar, 2014.
[308] M. H. P. C. B. V. Rao, "EMG signal based finger movement recognition for prosthetic hand control," presented at the Communication, Control and Intelligent Systems (CCIS), 2015, Mathura, India, 2015.
[309] G. R. N. a. H. T. Nguyen, "Nonnegative Matrix Factorization for the Identification of EMG Finger Movements: Evaluation Using Matrix Analysis," *IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS,* vol. 19, no. 2, pp. 478-485, 2015.
[310] S. H. Rocio Alba-Flores, and A. Shawn Mirzakani, "Performance Analysis of Two ANN Based Classifiers for EMG Signals to Identify Hand Motions," presented at the SoutheastCon, 2016, Norfolk, VA, USA, 2016.
[311] A. H. A.-T. Rami N. Khushaba, Ahmed Al-Ani, and Adel Al-Jumaily, "A Framework of Temporal-Spatial Descriptors based Feature Extraction for Improved Myoelectric Pattern Recognition," *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* vol. PP, no. 99, 2016.
[312] D. S. K. Minjae Kim, and Wan Kyun Chung, "Microneedle-Based High-Density Surface EMG Interface with High Selectivity for Finger Movement Recognition," presented at the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 2016.
[313] J. K. Ashnil Kumar, David Lyndon, Michael Fulham, and Dagan Feng, "An Ensemble of Fine-Tuned Convolutional Neural Networks for Medical Image Classification," *IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS,* vol. 21, no. 1, pp. 31-40, 2016.
[314] Y. Z. Ruikai Zhang, Tony Wing Chung Mak, Ruoxi Yu, Sunny H. Wong, James Y. W. Lau, and Carmen C. Y. Poon, "Automatic Detection and Classification of Colorectal Polyps by Transferring Low-Level CNN Features From Nonmedical Domain," *IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS,* vol. 21, no. 1, pp. 41 - 47 2016.
[315] A. G. Karim Lekadir, A` ngels Betriu, Maria del Mar Vila, Laura Igual, Daniel L. Rubin, Elvira Fern´andez, Petia Radeva, and Sandy Napel, "A Convolutional Neural Network for Automatic Characterization of Plaque Composition in Carotid Ultrasound," *IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS,* vol. 21, no. 1, pp. 48 - 55, 2016.
[316] M. A. Stergios Christodoulidis, Lukas Ebner, Andreas Christe, and Stavroula Mougiakakou, "Multisource Transfer Learning With Convolutional Neural Networks for Lung Pattern Analysis," *IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS,* vol. 21, no. 1, pp. 76 - 84 2016.
[317] J. S. Liu and Y. N. Wu, "Parameter expansion for data augmentation," *Journal of the American Statistical Association,* vol. 94, no. 448, pp. 1264-1274, 1999.
[318] M. A. Tanner and W. H. Wong, "The calculation of posterior distributions by data augmentation," *Journal of the American statistical Association,* vol. 82, no. 398, pp. 528-540, 1987.
[319] I. Cortes-Ciriano and A. Bender, "Improved Chemical Structure–Activity Modeling Through Data Augmentation," *Journal of chemical information and modeling,* vol. 55, no. 12, pp. 2682-2692, 2015.
[320] S. Hauberg, O. Freifeld, A. B. L. Larsen, J. Fisher, and L. Hansen, "Dreaming more data: Class-dependent distributions over diffeomorphisms for learned data augmentation," in *Artificial Intelligence and Statistics*, 2016, pp. 342-350.

[321] J. Ahmad, K. Muhammad, and S. W. Baik, "Data augmentation-assisted deep learning of hand-drawn partially colored sketches for visual search," *PloS one,* vol. 12, no. 8, p. e0183838, 2017.
[322] J. Fintzi, X. Cui, J. Wakefield, and V. N. Minin, "Efficient Data Augmentation for Fitting Stochastic Epidemic Models to Prevalence Data," *Journal of Computational and Graphical Statistics,* vol. 26, no. 4, pp. 918-929, 2017/10/02 2017.
[323] X. Cui, V. Goel, and B. Kingsbury, "Data augmentation for deep neural network acoustic modeling," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP),* vol. 23, no. 9, pp. 1469-1477, 2015.
[324] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, 2017, pp. 5220-5224: IEEE.
[325] Z. Gan, R. Henao, D. Carlson, and L. Carin, "Learning deep sigmoid belief networks with data augmentation," in *Artificial Intelligence and Statistics*, 2015, pp. 268-276.
[326] G. Rogez and C. Schmid, "MoCap-guided data augmentation for 3D pose estimation in the wild," in *Advances in Neural Information Processing Systems*, 2016, pp. 3108-3116.
[327] D. L. Piza, A. Schulze-Bonhage, T. Stieglitz, J. Jacobs, and M. Dümpelmann, "Depuration, augmentation and balancing of training data for supervised learning based detectors of EEG patterns," in *Neural Engineering (NER), 2017 8th International IEEE/EMBS Conference on*, 2017, pp. 497-500: IEEE.
[328] A. Boyajian, "The tensor-a new engineering tool," *Transactions of the American Institute of Electrical Engineers,* vol. 55, no. 8, pp. 856-862, 1936.
[329] J. Betten, "Applications of tensor functions in continuum damage mechanics," *International Journal of Damage Mechanics,* vol. 1, no. 1, pp. 47-59, 1992.
[330] A. Bernardi, J. Brachat, P. Comon, and B. Mourrain, "General tensor decomposition, moment matrices and applications," *Journal of Symbolic Computation,* vol. 52, pp. 51-71, 2013.
[331] D. Yu, L. Deng, and F. Seide, "The deep tensor neural network with applications to large vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 21, no. 2, pp. 388-396, 2013.
[332] M. Sørensen and P. Comon, "Tensor decompositions with banded matrix factors," *Linear Algebra and its Applications,* vol. 438, no. 2, pp. 919-941, 2013.
[333] J. He and T.-Z. Huang, "Inequalities for M-tensors," *Journal of Inequalities and Applications,* vol. 2014, no. 1, p. 114, 2014.
[334] S. Puri, S. J. Pandey, G. Bhat, and N. Raina, "Estimation of Source Parameters and Moment Tensor Solution through Waveform Modeling of 2013 Kishtwar Earthquake," *World Academy of Science, Engineering and Technology, International Journal of Geological and Environmental Engineering,* vol. 4, no. 4, 2017.
[335] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky, "Tensor decompositions for learning latent variable models," *Journal of Machine Learning Research,* vol. 15, pp. 2773-2832, 2014.
[336] A. Cichocki *et al.*, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Processing Magazine,* vol. 32, no. 2, pp. 145-163, 2015.
[337] V. Hore *et al.*, "Tensor decomposition for multiple-tissue gene expression experiments," *Nature genetics,* vol. 48, no. 9, pp. 1094-1100, 2016.