



YAŞAR UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

PHD THESIS

**ONLINE ADAPTIVE CLASSIFICATION OF FINGER
MOVEMENTS FOR BRAIN COMPUTER INTERFACES**

MOHAND LOKMAN AHMAND AL-DABAG

THESIS ADVISOR: ASST. PROF. DR. NALAN ÖZKURT

COMPUTER ENGINEERING

PRESENTATION DATE: 11.04.2019

BORNOVA / İZMİR
April 2019

We certify that, as the jury, we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of the Doctor of Philosophy.

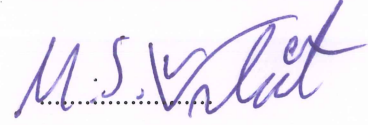
Jury Members:

Signature:

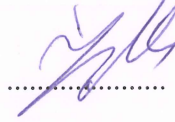
Prof. Dr. Mehmet Kuntalp
Dokuz Eylül University



Prof. Dr. Mehmet Süleyman Ünlütürk
Yasar University



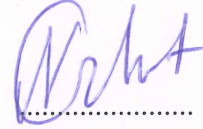
Asst. Prof. Dr. Yalçın İşler
Katip Celebi University



Asst. Prof. Dr. Korhan Karabulut
Yasar University



Asst. Prof. Dr. Nalan Özkurt
Yasar University



Prof. Dr. Cüneyt GÜZELİŞ
Director of the Graduate School

ABSTRACT

ONLINE ADAPTIVE CLASSIFICATION OF FINGER MOVEMENTS FOR BRAIN COMPUTER INTERFACES

Al-dabag, Mohand

PHD, Computer Engineering

Advisor: Dr. Nalan Özkurt

April 2019

Studies on Brain computer interface (BCI) are challenging and promising since it establishes an indirect link between human and machine. This link depends on a correct interpretation of the brain activities to control the machine based usually on noninvasive brain acquisition system which is called Electroencephalography (EEG). Since EEG signals record the brain cortical activities indirectly through using electrodes placed on the scalp of the head, is seriously affected by different sources of noises such as bad electrode connection, power line.. etc. This thesis construct an online adaptive BCI system based on EMOTIV EPOC+ headset using MATLAB and C# programming languages. The study first focused on the offline study to propose an algorithm, which constructs discriminative cognitive features to classify the right/left fingers EEG movement signals. After preprocessing and denoising the EEG signals, the proposed method uses two classifiers (Multilayers perceptron network MLP and support vector machine SVM) to classify the features produced from ten statistical moments of cross-correlated channels. It is observed that cross-correlation of effective channels with right and left hemisphere channels constructs more discriminative features from the EEG signals. The classification results show that the two classifiers have competitive results (96% on average) to classify 13 subjects offline dataset. Also these results show, SVM is more effective than MLP in term of computation time. Genetic algorithm is also used to select the best features and it is found that only three statistics (mode, maximum, and standard deviation) are enough to discriminate the cognitive EEG signals. After feature selection, SVM classifier still has competitive classification rates with MLP and faster computation time than MLP. Therefore only, SVM classifier was used in the online phase. Secondly, an online BCI software is

constructed using by imbedding the MATLAB algorithm in C# BCI platform to make use of the scientific and parallelism facilities of both programming languages. This online system is based on the enhanced method proposed in the offline phase. It used a very common solution to synchronizing the concurrent processing (producer/consumer problem) for simulating the functionality of the two stage pipeline system. This simulation provides a real time acquisition of EEG motor signal to the preprocessing and classification stage and provides an interactive online response time rather than the sequential processing system.

Keywords: Brain computer interface, pipeline, cross-correlation, discrete wavelet transform (DWT), genetic algorithm.



ÖZET

BEYİN BİLGİSAYAR ARAYÜZLERİ İÇİN EL HAREKETLERİNİN ÇEVİRİMİÇİ UYARLANIR MODELLENMESİ

Al-dabag, Mohand

PHD, Computer Engineering

Advisor: Dr. Nalan Özkurt

April 2019

Beyin bilgisayar arayüzü (BBA) çalışmaları insan ile makine arasında dolaylı bir bağlantı kurduğu için hem zorludur hem de ümit vadeder. Bu bağlantı genellikle girişimsel olmayan beyin sinyalleri algılama sistemi Elektroensefalografi (EEG) sinyallerinin makineyi yönetmek için doğru şekilde yorumlanması ile kurulur. EEG sinyalleri, beyin korteksindeki elektriksel aktiviteyi kafa derisi üzerinden dolaylı olarak kaydettiği için EEG sinyalleri, kötü kontak, güç hattı gürültüleri vb. gibi çok büyük miktarda bozulmadan mustarıptır. Bu tezde, MATLAB ve C# programlama dilleri kullanılarak EMOTIV EPOC+ EEG cihazı tabanlı bir gerçek zamanlı uyarlanı BBA sistemi gerçekleştirilmiştir. Tezde önce EEG hareket sinyallerinden sağ ve sol parmak hareketlerinin ayrıştırılması için çevrimdışı bir algoritma geliştirilmesine odaklanılmıştır. Önışleme ve gürültü azaltmanın ardından, sistem kanalların çapraz ilintilerinin istatistiksel momentlerini sınıflandırmak için Çok Katmanlı Yapay Sinir Ağı (YSA) ve Destek Vektör Makineleri (DVM) kullanır. Etkin kanalların sağ ve sol lob kanallarıyla ilintileri daha ayırıcı öznelikler oluşturduğu gözlenmiştir. Sonuçlar YSA ve DVM'nin ortalamada %96 ile benzer başarıya sahip olduğunu gösterirken, DVM her zaman daha hızlıdır. Genetik algoritma ile istatistiksel özneliklerden en ayırıcılar arandığında, parmak EEG sinyallerinin ayrılabilmesi için mod, en büyük ve standart sapmanın yeterli olduğu görülmüştür. Öznelik seçiminden sonra da DVM, YSA ile benzer sınıflandırma oranlarına sahipken daha hızlı sonuç vermektedir. İkinci olarak MATLAB ve C# programlama dillerinin sırasıyla bilimsellik ve paralellik özellikleri kullanılarak gerçek zamanlı BBA yazılımı gerçekleştirilmiştir. Gerçek zamanlı sistem, çevrimdışı sistemde geliştirilen önışleme ve sınıflandırma yöntemlerini kullanarak kullanıcı arayüzü geliştirir. Bu sistem eş zamanlı işleme (üretici/tüketici) probleminin yaygın bir çözümünü iki aşamalı ardışık düzen benzetiminde kullanır. Bu

da EEG hareket sinyallerinin gerek zamanlı kaydı ve işlenmesi/sınıflandırılmasını ardışıl yerine interaktif çevrimii tepki zamanıyla işlenmesini sağlar.

Anahtar Kelimeler: Beyin bilgisayar arayüzü, Elektroansefalografi, ardışık düzen işleme, apraz-ilinti, genetik algoritma.



ACKNOWLEDGEMENTS

I would first like to thank and appreciate my thesis advisor Dr. Nalan Özkurt for her guidance and infinite support during this study especially for helping me to find the EEG dataset participants. Since I am a foreigner student, she contacted with her friends and students to find participants for recording the EEG offline and online datasets. She even helped me in some sessions to acquire the dataset. I really thank her kindness and generosity for guiding me to finish this work. She hasn't spared any effort to support me and she provided me all her knowledge and experience to complete this work. I also appreciate her contributions for helping me to publish our first article in the science citation index journal and her supports for participating in two other conferences.

I owe my sincere gratitude to the members of the progressing Committee for helping us to finish this work. Their valuable comments and advices added a lot of scientific sobriety to this work and led us in the right way towards ending this work correctly.

I also want to express my frank gratitude to my neighbor Mr Ahmet Ozman and his family for their friendship. They support me a lot and act like my second family in Turkey. Mr Ahmet always asked for me and even he helped me to solve some problems.

I express my thankfulness for Mr. Huseyin Can and his family. They really made me shy with their invitations generosity which provided great emotional support for me especially because my family is living abroad. I would like to express my full gratitude to my big brother Mr. Huseyin for his helpfulness to solve my normal living problems like renting a flat, finding an internet company and even go to the hospital with me.

I also kindly appreciate all efforts of the participants for helping us to acquire the offline and online dataset especially the participated lecturers of Yasar University. They spent their valuable time and bore all boring acquisition procedure to help us for recording the EEG datasets.

Finally, I would like to thank my dearest wife for supporting and giving me all the motivations to finish this work. She had been taking care of my family and children throughout my studies, and she did all my duties like as I were with them. Therefore, I really appreciate her efforts.

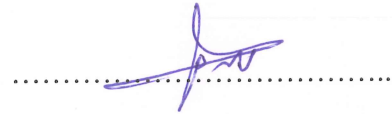
Mohand Lokman Al-dabag
İzmir, 2019

TEXT OF OATH

I declare and honestly confirm that my study, titled “ONLINE ADAPTIVE CLASSIFICATION OF FINGER MOVEMENTS FOR BRAIN COMPUTER INTERFACES” and presented as a PhD Thesis, has been written without applying to any assistance inconsistent with scientific ethics and traditions. I declare, to the best of my knowledge and belief, that all content and ideas drawn directly or indirectly from external sources are indicated in the text and listed in the list of references.

Mohand Lokman Ahmad Al-dabag

Signature



April 11,2019

TABLE OF CONTENTS

Abstract.....	ii
Özet.....	iv
BEYİN BİLGİSAYAR ARAYÜZLERİ İÇİN EL HAREKETLERİNİN ÇEVİRİMİÇİ UYARLANIR MODELLENMESİ	iv
ACKNOWLEDGEMENTS.....	vi
TEXT OF OATH.....	vii
TABLE OF CONTENTS	viii
LIST OF FIGURES	xii
LIST OF TABLE.....	xvi
SYMBOLS AND ABBREVIATIONS.....	xvii
CHAPTER 1 Introduction	1
1.1 Brain Computer Interfaces	1
1.2 Literature Review	1
1.3 Aim of the Thesis	4
1.4 Contributions	5
1.5 Thesis Outline.....	6
CHAPTER 2 EEG signal.....	8
2.1 Introduction	8
2.2 Human brain anatomy	8
2.3 Action Potentials	10
2.4 EEG Generation.....	10
2.5 Brain Rhythms.....	11
2.6 EEG measurement	13
2.7 Electrode Placement	13
2.8 Montages	14
2.9 EMOTIV headset	15
CHAPTER 3 Data Analysis.....	17
3.1 Introduction	17
3.2 Wavelet Transform.....	18

3.2.1 Continuous Wavelet Transform.....	19
3.2.2 Inverse WT.....	21
3.2.3 Discrete Wavelet Transform.....	22
3.2.4 Dyadic Grid Scaling of Wavelet Transforms	22
3.2.5 Scaling Function and Multiresolution Representation.....	23
3.2.6 Fast Wavelet Transform FWT	24
3.2.7 Thresholding and Denoising	26
3.3 Feature Extraction	27
3.3.1 Correlation	28
3.3.2 Measuring the Central Tendency	28
3.3.3 Measuring the Dispersion of Data	29
3.4 Feature Selection	29
3.4.1 Genetic Algorithms (GA)	30
CHAPTER 4 Machine Learning and Parallel Processing	34
4.1 Introduction	34
4.2 Machine Learning Algorithms for Classification	34
4.2.1 Measuring the Classifier Accuracy.....	34
4.2.2 Evaluating the Performance of a Classifier	35
4.2.3 Multilayer Neural Networks	37
4.2.4 Support Vector Machines	39
4.3 Basic Pipeline Concepts	42
4.4 Concurrency and semaphore	43
4.4.1 The Producer/Consumer Problem.....	43
CHAPTER 5 offline methology and results	45
5.1 Introduction	45
5.2 EEG Datasets.....	46
5.2.1 EMOTIV Dataset.....	46
5.2.2 BCI Competition III Data set IVa.....	47

5.3 Classifiers Settings	48
5.4 Proposed Method.....	48
5.4.1 Experiments and Results	50
5.4.2 Patient-Based Classification	50
5.4.3 Movement Based Classification.....	54
5.4.4 Comparison with Previous Studies	56
5.5 The Enhanced Method.....	57
5.5.1 Experiments and Results	60
5.6 Feature selection and classification	63
CHAPTER 6 online methology and results.....	69
6.1 Introduction	69
6.2 Experimental Procedure	69
6.3 Constructing BCI platform.....	70
6.4 EMOTIV EPOC+ Configuration	71
6.5 GUI for Online BCI.....	72
6.5.1 Signal save panel.....	73
6.5.2 Task event panel.....	73
6.5.3 EEG signal panel.....	73
6.5.4 Recording and training panel	74
6.5.5 Testing panel	75
6.6 Processing time.....	76
6.7 Classification rates	77
6.8 Discussions	79
CHAPTER 7 conclusion and future work	82
7.1 Summary.....	82
7.2 Conclusions	82
7.3 Future Works	84
REFERENCES	85



LIST OF FIGURES

Figure 2.1: The Basic Brain Anatomy.....	8
Figure 2.2: Neuron Structure (Sanei & Chambers, 2007).....	10
Figure 2.3: An Example of Action Potential.....	10
Figure 2.4: The Three Main Layers of The Brain And Their Approximate Resistivities And Thicknesses (Ω =ohm).....	11
Figure 2.5: The 10-20 System Electrode Placement (Grainann, Allison, & Pfurtscheller, 2010).....	14
Figure 2.6: Electrodes of EMOTIV EPOC+ Headset.....	15
Figure 3.1: Fourier Transform Convert The Signal to Various Sinusoids of Different Magnitudes And Frequencies.....	18
Figure 3.2: The Windowing Comparison Between Three Transformation (FFT, STFT, WT (Misiti, Misiti, Oppenheim, & Poggi (2002))...)	18
Figure 3.3: Dilation Factor A ($a_1 = a_2/2$; $a_3 = 2a_2$) of The Wavelet Function.	19
Figure 3.4 : Wavelet Translation.....	20
Figure 3.5: Wavelet Transform of Sinusoidal Signal.....	21
Figure 3.6: Single Level Fast Wavelet Transform (Burrus, Gopinath, & Guo, 1998).....	25
Figure 3.7: Two Stage Decomposition of FWT (Burrus, Gopinath, & Guo, 1998).....	25
Figure 3.8: Two Stages of FW Synthesis (Burrus, Gopinath, & Guo, 1998)..	26
Figure 3.9: Schematical Behavior of Magnitude Thresholding.....	26
Figure 3.10: Relationship Between The Original Wavelet Coefficient And The Thresholded Coefficients.....	27
Figure 3.11: Steps of Genetic Algorithm (Talbi, 2009).....	31
Figure 3.12 : Roulette Wheel Selection (Talbi, 2009).....	32
Figure 3.13: Tournament Selection (Talbi, 2009).....	32
Figure 3.14: One Point Crossover (Michael, 2005).....	33
Figure 3.15 : Mutation Operation (Michael, 2005).....	33
Figure 4.1: Holdout Method for Estimating Classifier Accuracy (Jiawei & Micheline, 2006).....	36

Figure 4.2: Mulilayer Perceptron with Hidden Layers.....	37
Figure 4.3: Three layer backpropagation neural network.	38
Figure 4.4: The SVM for The Linearly Separable Case.....	40
Figure 4.5: The Concept of The Slack Parameter.	41
Figure 4.6: Pipeline Concepts (a) Pipeline Timing Diagram (b) Non Pipeline Timing Diagram.	42
Figure 4.7: Producer/Consumer Pseudo Code, (a) Producer Code (b) Consumer Code.	44
Figure 5.1: EMOTIV Dataset Real Finger Movements.	46
Figure 5.2: A Sample of EMOTIV Dataset.....	47
Figure 5.3: A Sample of BCI Competition III Dataset IVa	48
Figure 5.4: The Selected EEG Motor Channels for The Both Datasets.....	49
Figure 5.5: The Proposed Method Structure.	50
Figure 5.6: Patient Based Classification Experiments to Determine The Number of Nodes in The Hidden Layer for BCI Competition Dataset.....	51
Figure 5.7: Patient Based Classification of BCI Competition III VIa Dataset.	52
Figure 5.8 : Patient Based Classification Experiments to Determine The Number of Nodes in The Hidden Layer for EMOTIV Dataset.	52
Figure 5.9 : Patient Based Classification Of EMOTIV Dataset.....	53
Figure 5.10: The Average of All Subject Accuracies in Both Datasets for Both Classifiers.	53
Figure 5.11 : Movement Based Classification Experiments to Determine The Number of Nodes in The Hidden Layer for BCI Competition Dataset.....	54
Figure 5.12: Movement Based Classification Experiments to Determine The Number of The Nodes in The Hidden Layer for EMOTIV Dataset.	55
Figure 5.13: The Performance of Both Classifier Using Movement Based Classification.....	55
Figure 5.14: The Enhanced Method Block Diagram.....	58

Figure 5.15 : The Stages Performance of The Enhanced Method on FC5 Channel for Both Tasks (Right/Left Fingure Movement).....	60
Figure 5.16: EEG Topography of Alpha Band Before And After Applying Eeg Artifacts Removal. (a) Original EEG Right Movement. (b) Denoised EEG Right Movement. (c) Original EEG Left movement. (d) Denoised EEG Left Movement.	61
Figure 5.17: Effects of Cross Correlation Stage with Effective Channels. (a) Right Movements Channels Before Cross Correlation. (b) Right Movements Channels After Cross Correlation. (c) Left Movements Channels Before Cross Correlation. (d) Left Movements Channels After Cross Correlation.	61
Figure 5.18: MLP Performance Evaluation with Different Number of Nodes in The Hidden Layer.....	62
Figure 5.19: Classification Rates of Both Classifiers for All Subjects.	63
Figure 5.20: The Average Classification Rates of Both Classifiers for All Subjects.....	63
Figure 5.21: SVM Classification Rates Before And After Feature Selection for 13 Subjects.	64
Figure 5.22: MLP Classification Rates Before And After Features Selection.	65
Figure 5.23: MLP Classification Rates with SD Using Different Number of Neurons in The Hidden Layer.	65
Figure 5.24: MLP Performance Using Two Different Architectures.	66
Figure 5.25: The Average of SVM Classification Rates Before And After Features Selection.....	66
Figure 5.26: The Average of MLP Classification Rates Before And After Feature Selection of Two Different MLP Structure.....	67
Figure 5.27: Processing Time of SVM Using Two Different Number of Statistical Methods.	67
Figure 5.28: MLP Processing Time of Different Architectures.	68
Figure 6.1: Finger Movement Procedure.....	70
Figure 6.2 : EMOTIV Cortex UI Platform.	72

Figure 6.3: Real Time Software Platform And Its Panels.	72
Figure 6.4: EEG Viewer Window.	74
Figure 6.5: Thesis Software on Action.	75
Figure 6.6: The Pipeline Structure.	76
Figure 6.7: The Processing Time of The Pipeline Stage Two.	77
Figure 6.8: Real Time Classification Rates of Five Subjects (S1-S5).	78
Figure 6.9: Response Time of Nonpipelined And Pipelined System.	79
Figure 6.10: Theoretical And Simulation Processing Time of Different Shifting Percentage.	80



LIST OF TABLE

Table 2.1: Functions of Lobes within Cerebral Hemispheres	9
Table 4.1: A Confusion Matrix for A Binary Classification Problem.	35
Table 5.1 : Performance of Several Methods for Classifying Imaginary Motor EEG of BCI Competition III Dataset IVa.	56
Table 5.2: EEG Features Ratios Before And After Features Selection.....	64
Table 6.1: Performance Comparison with Other Studies	78
Table 6.2: Shifting Percentage And Its Number Of Trials	80



SYMBOLS AND ABBREVIATIONS

ABBREVIATIONS:

PCA	Principal Component Analysis
BSS	Blind Source Separation
ICA	Independent Component Analysis
CSP	Common Special Pattern
CNS	Central Nervous System
AP	Action Potential
MEG	Magnetoencephalography
EMG	Electromyogram
ECG	Electrocardiogram
EEG	Electroencephalogram
EOG	Electroocclugram OR electrooptigram
EKG	Electrogastrogram
PET	Positron Emission Tomography
fMRI	Functional Magnetic Resonance Imaging
BCI	Brain Computer Interface
FFT	Fast Fourier Transform
WT	Wavelet Transform
STFT	Short-Time Fourier Transform
DWT	Discrete Wavelet Transform
GA	Genetic Algorithms
MLP	Multilayer Perceptron
SVM	Support Vector Machines
SD	Standard Deviation

SYMBOLS:

δ	Delta rhythm
θ	Theta rhythm
α	Alpha rhythm
β	Beta rhythm
γ	Gamma rhythm
ψ	Wavelet function
ϕ	Scaling function
σ	Standard deviation

CHAPTER 1

INTRODUCTION

1.1 Brain Computer Interfaces

Brain Computer Interface (BCI) systems are the systems that allow humans to communicate with a machine through brain signals and they have diverse applications, such as smart living, entertainment, and neuro prostheses. BCIs also provide a new chance to rehabilitate disabled people by making them able to interact with their environment without the help of their family members. To acquire brain signals, Electroencephalography (EEG) signal is widely used as a noninvasive system. It records the cerebral cortex electrical activity of the brain through a number of electrodes on the scalp. Using EEG allows us to detect motor movement and motor imagery which is the mental rehearsal of a movement (such as a hand or foot movement) without executing it, and it can be used as a control signal in BCI systems.

The EEG devices used for medical diagnosis are complex devices with a large number of electrodes and not comfortable especially during sleep EEG measurements. For a BCI system designed for everyday use of a disabled person or for playing games, the ergonomic system will be an important aspect as well as the accuracy. With the advances in sensor and communication technology, now it is possible to design wireless EEG devices with acceptable signal resolution and accuracy.

In the following section, challenges in EEG processing and the BCI studies in literature will be briefly reviewed.

1.2 Literature Review

EEG classification is still confronting serious challenges for many reasons. Firstly, the EEG signals are collected indirectly from human scalp, so they are distorted signals of the actual brain signals. Secondly, the artifacts and noises introduced by the body

interfere with EEG signals, all these weaken the EEG signal and make it hard to be classified. The examples of those contaminations are measurement instruments noise and bad electrode connections (Rakendu & Reza, 2005). The human body also adds some artifacts to EEG signals. These artifacts could be caused by eye blinks or eye movements, and both are called ocular artifacts (OA) or electrooculography (EOG) artifacts which mainly and frequently interfere with EEG signal in anterior scalp regions. Electromyogram (EMG), caused by muscles movements, could be another source of artifacts that weaken EEG signals. EOG and EMG occur frequently with amplitude several times larger than EEG signals. Therefore, these artifacts seriously confuse the interpretation of EEG signals. Several techniques were suggested to remove these artifacts from EEG signals (Rakendu & Reza, 2005) (Junfeng, Pan, Yong, Pei, & Chongxun, 2010).

The simplest technique to reject the EEG segments contaminated by artifacts is done by inspection the EEG data visually. However, this means losing the whole contaminated EEG data and this is unacceptable especially when the artifacts occur frequently (Junfeng, Pan, Yong, Pei, & Chongxun, 2010). Some methods were proposed for removing EOG based on the time domain or frequency domain regression. Reliable reference EOG channel is always needed in these methods but this channel is also corrupted with EEG signal (Manish, Rohan, M., & Ajoy, 2017). Therefore, these methods may not be suitable for EOG removal (Junfeng, Pan, Yong, Pei, & Chongxun, 2010). Other methods proposed to remove these artifacts are based on the Principal component analysis (PCA) or based on the independent component analysis (ICA) (Junfeng, Pan, Yong, Pei, & Chongxun, 2010) (Manish, Rohan, M., & Ajoy, 2017) (Sim, et al., August 2017). Another technique depends on time amplitude thresholding (Manish, Rohan, M., & Ajoy, 2017).

Different schemes are suggested to classify EEG motor signals due to the complexity of the waveform. While some of the studies focus on separation of EEG rhythm bands, the others make classification by focusing on motor EEG channels. There were also some studies tried to extract the original motor signal and eliminate artifacts by using blind source separation (BSS) methods, such as Principal Component Analysis (PCA), Independent Component Analysis (ICA). These are complex algorithms which need excessive computation time to find the source signal. Other studies try to find a

template signal between the training set like Common Special Pattern (CSP). Some of the studies are considered as follows:

The first study applied two search algorithms (metagene (AM) algorithm and the Bat optimization algorithm (BA)) to search for the most discriminative common special pattern CSP feature to optimize the SVM parameters (Selim, Tantawi, Shedeed, & Badr, 2018). The other study searched for an optimal feature subset using a Differential Evolution (DE) optimization algorithm. They used filtering to extract μ and β bands, CSP and Support Vector Machine (SVM) as a classifier. They had in average 95% classification accuracy with a minimum of just 10 features (Muhammad, Nauman, Hubert, & Shum, 2017). The next study extracts three different feature groups (time, frequency and time-frequency features) and used five Evolutionary Computation (EC) algorithms to find best features with an ANN as a classifier (Bahareh, Mohammad, Dian., & Vinod, 2017). Other researchers use ICA and four well-known features (adaptive autoregressive (AAR) parameters, and Hjorth parameters, power spectral density (PSD) and discrete wavelet coefficients). Firefly algorithm and Self-Adaptive firefly algorithm are used for electrodes and features selection with SVM classifier to have good classification rates (Rimita, Pratyusha, & Amit, 2017).

There are also some methods, which extracts the feature using extensive computation. In the first study, the researchers applied one dimension-aggregate approximation (1d-AX) to construct an effective signal representation for long short-term memory (LSTM) networks and channel weighting technique is further applied to enhance the classification rates (Wang, Jiang, Liu, Shang, & Zhang, 2018). Another study deployed an artifact rejected CSP (AR-CSP) for extracting motor imaginary MI features to the neuro-fussy classifier. This is performed to develop the MI EEG-based BCIs (Jafarifarmand, Badamchizadeh, Khanmohammadi, Nazari, & Tazehkand, 2018). The next researchers retained the temporal features by performing the multi-channel series in CSP space. Then they encoded the multi-channel data by applying the separated channel convolutional network to extract features for a recognition network (Zhu, et al., 2019).

Another important BCI topic is to investigate the effective motor channel(s). The researchers performed the sequential forward search method for each subject to find the effective channels that maximize the SVM classification rates. This was done after

preprocessing the EEG signal using a moving average filter and extraction of some features using the Hilbert transform (Aydemir & Ergün, 2019). The other study performed filtering techniques to extract alpha and beta bands from channels (C1,C3,C5,CP3). Then, they used Linear Discriminant Analysis (LDA) to distinguish the presence/absence of real right hand movement. (McCrimmon, et al., 2017). The next study applied band pass filter onto (C4 and C3) channels to construct beta rhythm. Then, they evaluated the frequency band energy, Hilbert transform and Phase Locking Value (PLV) to extract features for support vector machine (SVM) (Liu, et al., 2017).

The online BCI is still an open research subject because each part of the system has effects influencing the overall system performance. In one of the studies, it is tried to control the movement of an online robot arm using five motor imagery mental commands acquired via EMOTIV EPOC+ headset. They performed three days of offline training using 5-fold cross validation to train the system and then performed the online testing. The classification was accomplished by an Adaptive Neural Fuzzy Inference System and the accuracy rate is measured as 65-70% (Bhattacharyya, Basu, & Amit Konara, 2015). Another study used an open source software (Open ViBE) to acquire EEG signals from EMOTIV EPOC+ headset and makes online/offline classification using Support Vector Machine (SVM). The average recognition rate in offline testing and single trial classification is 60.63% for right arrow and 45.93% for left arrow (Risangtuni, Suprijanto, & Widyotriatmo, 2012). Next study proposed a method for controlling the Arduino LED via eye blink. They used Simulink for communication between the Arduino and the processed EEG data stored in Matlab workspace (Mahajan & Bansal, 2017). Another study implemented an online BCI system using OpenBCI headset and FPGA. The online classification performance of this system was about 75% on average (Belwafi, Romain, Ghaffari, Djemal, & Ouni, 2018).

1.3 Aim of the Thesis

The researchers have been interested to develop some techniques for improving the BCI operation. In recent years, this interest is accelerated due to the need of installing new channel between human and machine. This channel not only improves the lives

of the disabled people but also improves the life quality of healthy people. As described in the literature review, there are still challenges to overcome.

The objective of this thesis is to design an online adaptive mobile BCI system. Since EEG signal characteristics change from one person to another, an adaptive system is aimed to fit the needs of different users. Furthermore, the goal of the mobile system is to provide a comfortable system for the user without cables around.

Therefore, this thesis addresses the following issues:

- Developing an accurate and fast algorithm for preprocessing, feature extraction and classification by using pre-recorded data which is called the offline algorithm later in this thesis;
- Constructing an online upgradable BCI platform based on EMOTIV EPOC+ headset. This platform uses the best and the fastest algorithm provided by the offline study.

The scope of the study is limited to classification of left and right finger movement in order to demonstrate the applicability of the algorithm in both offline and online cases.

1.4 Contributions

In the offline study, an EEG signal processing and classification system is proposed to classify the EEG motor signals. The preprocessing stage is specialized for motor movement classification. The cross-correlation of effective channels with right and left hemisphere channels increases the discriminative abilities of the extracted features. The proposed algorithm is based on relatively simple tools which is a crucial demand for the real time system. The algorithm simplicity leads to a quick response time which is demonstrated in the online processing. The online BCI also improved the performance of the EEG acquisition stage by simulating the pipeline mechanism which increased the system reliability and reduced the response time of the system. Thus the proposed system meets the aims of the thesis by providing a mobile system since it is based on wireless EEG headset and it also provides the ability to retrain the system which makes the system adaptive to different users.

1.5 Thesis Outline

Chapter 1 demonstrates the BCI system and provides a quick review of the challenges in design. It also presents the aim of the thesis and its novelty. Then, it explores the literature review, and finally a quick outline revision of the thesis is provided.

Chapter 2 demonstrates brief review about: the human brain anatomy, structure of neuron, the neuron action potential, the brain electrical activity and how EEG signal is generated in brain, the EEG rhythms, a description of the standard system that measures the EEG signal, the common montages to connect the EEG electrodes, and finally describes and lists the specification of EMOTIV EPOC+ headset that is used in this work.

Chapter 3 has two main topics. The first topic explores the signal analysis tools used in this thesis. It gives an explanation of the reasons for using wavelet transform and the reasons for preferring this transform. Then, a detailed description of wavelet transform is given. After that, it explains the common thresholding techniques used with the wavelet transform for denoising. The second topic provides a description of the statistical feature extraction methods used in this thesis. It also demonstrates the randomized search algorithm called genetic algorithm, and it provides the reason why this algorithm is preferable for feature selection.

Chapter 4 explores a summary in three issues. The first, it provides a brief description of the popular evaluation methods to evaluate the classifier performance. It also gives a brief explanation of two machine learning classifiers which are the SVM and multilayer perceptron network (MLP). Secondly, it expresses the pipelining technique used in parallel processing and its characteristics that distinguish it from traditional processing. Finally, it provides an abbreviation of concurrent processing and the solution of the most popular concurrent problem (producer/consumer problem).

Chapter 5 provides a detailed description about the two offline datasets used in this thesis and it also explains the setting of the two classifiers (MLP and SVM). Then, it provides a detailed explanation of the proposed method and its classification performance for both datasets. After that, it explores the reason for enhancing the

proposed method and explains the enhanced method with its impact on performance using only the EMOTIV dataset. It also lists the results after feature selection and the computation times of the two classifiers.

Chapter 6 explains the reasons for choosing the programming languages that are used to construct the online software. Then, a detail description is given for the online EMOTIV data acquisition procedure and it describes the way of converting the MATLAB program to a DLL file and how to link it with the C# program. After that, it demonstrates the online software platform, its components, and its facilities. An exploration is given to the computation times of the real time BCI and compares it with theoretical and simulation times. Finally, it demonstrates the classification rates of the online BCI software.

Chapter 7 expresses the conclusions that have been drawn from the offline/online classification rates and computation times. It also provides a suggestions for future works.

CHAPTER 2

EEG SIGNAL

2.1 Introduction

The nervous system is the neural network that forms and transmits information throughout the body in the form of electrical impulses. An electrical charge creates electrical energy, which has two important characteristics: it moves very quickly and in distinct “packets” called *impulses* (Norris & Siegfried, 2011)

Richard Caton (1842–1926), a scientist from Liverpool, England, is the first scientist that recorded the brain electrical activity by placing two electrodes of a galvanometer over the human scalp in 1875. Since then, the terminology for the electrical activity of the brain is electroencephalography (EEG) which is a combination of three terms: electro-(referring to the brain electrical activities acquisition) encephalo- (referring to the signals radiating from the head), and gram (or graphy), which means writing or drawing (Sanei & Chambers, 2007).

2.2 Human brain anatomy

The central nervous system (CNS) receives and integrates the information from sensory receptors and coordinates the body activity. The spinal cord and brain constitute the largest part of the nervous system.

Anatomically, the cerebrum, cerebellum, and brain stem are the three parts of the human brain, see Figure 2.1 (Norris & Siegfried, 2011) (Al-Aimama, 2013)

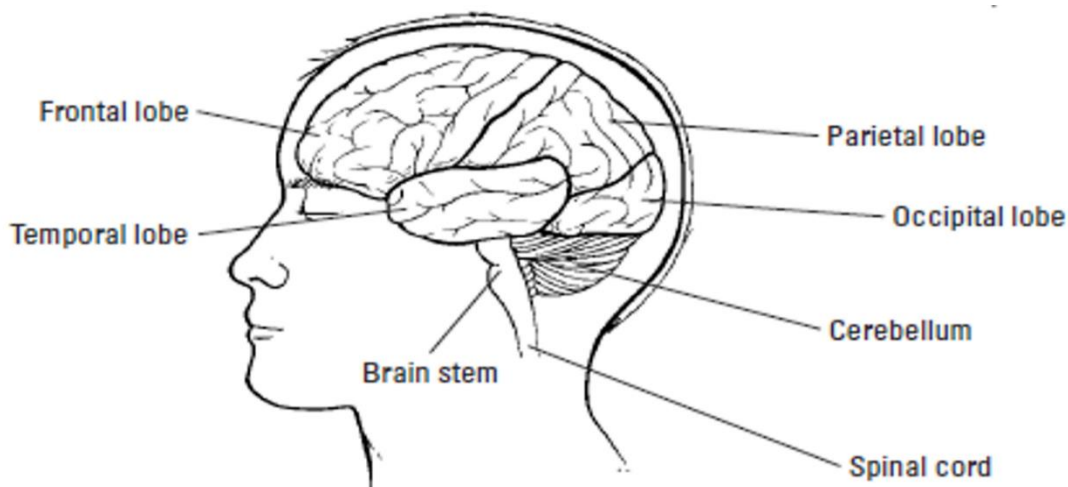


Figure 2.1: The Basic Brain Anatomy.

The cerebrum is divided into left and right halves which called the *left* and *right cerebral hemispheres*. The *frontal*, *parietal*, *temporal*, and *occipital* are four lobes of each brain hemisphere. The names of the lobes are derived from the skull bones that cover them. Table 2.1 lists each lobe functions (Norris & Siegfried, 2011)

Table 2.1: Functions of Lobes within Cerebral Hemispheres

Lobe	Functions
Frontal lobe	Voluntary muscle control, problem solving, concentration, planning, and speech production
Parietal lobe	Understanding speech, general interpretation area , ability to use words, sensations including heat/cold, pain, touch, and pressure
Temporal lobe	learning, Interpretation of sensations, hearing, remembering through sounds, and remembering visually
Occipital lobe	combining images received visually, recognizing objects visually, and Vision

The basic unit of the nervous system is the individual cell which is called a neuron. These neurons are specialized for the transmission and initiation of impulses (electrical signals). It has the ability to receive the outputs (pulses) of many other neurons, process it, decides whether it needs to initiate its own signal and pass it to other neurons, muscle, or gland cells.

Neurons have special cellular anatomy adapted to the quick transmission of an electrical charge. See Figure 2.2. All neurons have the same three parts, all enclosed within their cell membrane:

- **Cell body:** is similar to a generic cell and it contains the mitochondria, nucleus, and other organelles.
- **Dendrites:** are extensions that branch from one end of the cell body. They receive signals from other neurons and send electrical impulses towards the cell body.
- **Axon:** The axon transmits the impulses from the cell body to the next neuron forming a chain. (Think of electrical transmission wires.).It is like a cable on the opposite end of the dendrite (Norris & Siegfried, 2011).

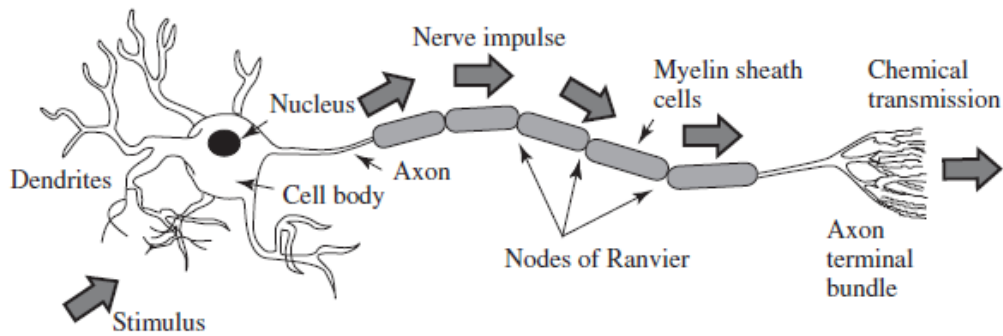


Figure 2.2: Neuron Structure (Sanei & Chambers, 2007).

2.3 Action Potentials

An Action Potential (AP) is an impulse that is emitted from a nerve to form a kind of information. APs are consequences of exchanging the ions across the neuron membrane and an AP is a temporary change in the membrane potential that is transmitted along the axon. It usually forwards in one direction and the cell body normally generated it. The membrane produces a spike by depolarizing the potential (becomes more positive). After reaching the peak of the spike, the membrane repolarizes (becomes more negative) and the potential becomes more negative than the resting potential and then returns to normal. The action potentials of most nerves last between 5 and 10 milliseconds. Figure 2.3 shows an example of AP (Nam, Nijholt, & Lotte, 2018) (Sanei & Chambers, 2007).

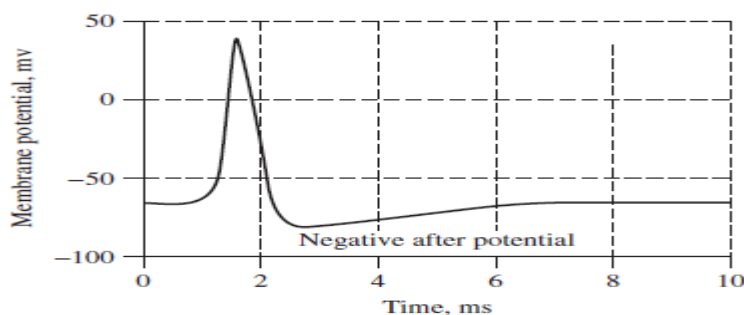


Figure 2.3: An Example of Action Potential.

2.4 EEG Generation

An EEG signal is a summation of the various neurons activities in the cerebral cortex and it measures the generated currents during the brain activities. The current generated by the synaptic excitation produces a magnetic field which is measurable by magnetic encephalography (MEG) machines and it is also produces a secondary

electrical field over the scalp measurable by EEG systems (Nam, Nijholt, & Lotte, 2018) (Sanei & Chambers, 2007).

The human head is constructed from different layers which include the scalp, skull, brain and many other thin layers in between, see Figure 2.4. The skull weakens the signals roughly one hundred times more than the soft tissue. However, most of the noise is generated either over the scalp (system noise or external noise) or within the brain (internal noise). Therefore, only huge populations of active neurons can generate satisfactory potential to be recordable using the scalp electrodes. These signals are later amplified greatly for display purposes.

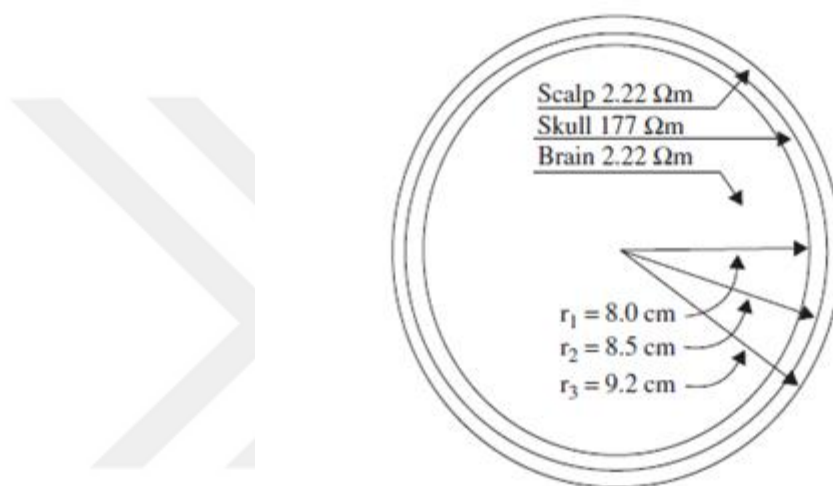


Figure 2.4: The Three Main Layers of The Brain And Their Approximate Resistivities And Thicknesses (Ω =ohm).

The acquired EEG signals from a human (and also from animals) may, for example, be used for investigation of the following clinical problems

1. Observation of brain death, coma, and alertness;
2. Localization of the damaged areas in the brain caused by head injury, stroke, and tumor;
3. Locating seizure origin and investigating epilepsy;
4. Testing epilepsy drug effects;
5. Investigating mental disorders; (Sanei & Chambers, 2007)

2.5 Brain Rhythms

Diagnosing of many brain disorders is done by the visual inspection of EEG signals and the clinical experts are familiar with monitoring of the brain rhythms in the EEG signals. The EEG signals frequencies and amplitudes change from one state to another

of the healthy adult, such as sleep and wakefulness. The features of the waves also vary with age. There are five main brain waves that can be distinguished by their frequency bands. Delta (δ), theta (θ), Alpha (α), beta (β), and gamma (γ) are these frequency bands from low to high frequencies respectively.

Delta (δ) band lies within 0.5–4 Hz range of the brain activities. The deep sleep is originally related to these rhythms and these rhythms may be appeared in the waking state. It is very easy to be corrupted with artifact signals due to the muscular movements of the jaw and neck.

The Theta waves frequencies lie within the range of 4–7.5 Hz. Theta waves appear as awareness slips towards drowsiness. The changes in the rhythm of theta waves are examined for maturational and emotional studies.

Alpha (α) waves are usually found over the occipital lobe of the brain and appear in the rear half of the head. They can be revealed in all parts of rear lobes of the brain. The frequency for alpha waves lies within the range of 8–13 Hz. Alpha waves exist in the motor movement and somatosensory and it has been thought it related to a relaxed awareness without any concentration or attention.

The frequency band within the range of 14–26 Hz of the brain electrical activity is called beta (β) wave. A beta wave is associated with active attention, focusing on the outside world, thinking, or problems solving, and all these activities are related to the waking status. Beta rhythm is mainly found over the central and frontal regions. The Beta band associated with Imaginary and real movement and presents a phenomenon known as event-related synchronization (ERS) and event-related desynchronization (ERD).

The gamma (γ) band are sometimes called the fast beta band and it exists above 30 Hz up to 45 Hz. Although this rhythm occurs rarely with very low amplitude but, the detection of these rhythms proofs the existence of certain brain diseases. The existing of this band proves to be a good indication of event-related synchronization (ERS) of the brain and can be used to indicate the presenting of right toes, right/left index finger movement, and tongue movement.

The above rhythms may last if the state of the subject does not change and therefore they are approximately cyclic in nature. On the other hand, there are other brain waveforms but the one that related to motor events is the mu rhythm (Sanei & Chambers, 2007) (Nam, Nijholt, & Lotte, 2018).

Mu (8–12 Hz) denotes motor and is strongly related to the motor cortex. Rolandic (central) mu is related to posterior alpha in terms of amplitude and frequency. From the mu rhythm the cortical functioning and the changes in brain (mostly bilateral) activities subject to physical and imaginary movements can be investigated (Sanei & Chambers, 2007) (J. R. Wolpaw, D. J. McFarland, & T. M. Vaughan, 2000).

2.6 EEG measurement

The early diagnosis of various human diseases need acquiring of signals and images from the human body. The electrobiological signals are one of these biological data which can be an electromyogram (EMG) from muscles, electrocardiogram (ECG) from the heart, magnetoencephalogram (MEG) or electroencephalogram (EEG) from the brain, electrooculogram (or electrooptigram, EOG) from eye nerves, and electrogastrogram (EGG) from the stomach (Sanei & Chambers, 2007).

The measurement of the brain activities has several methods such as Functional Magnetic Resonance Imaging (fMRI), Positron Emission Tomography (PET), and megnetoencephalography (MEG). However, EEG signal is still a useful tool for observing the brain activity because it is an inexpensive tool and it is also convenient for the patient (Yonghui Fang, Minyou Chen, & Xufei Zheng, 2015).

2.7 Electrode Placement

The differential amplifiers input signals are provided from the head via electrodes. They are attached to the head using a conductive silver chloride (AgCl) gel. The standard electrode placement guide is the 10-20 system of electrode placement which is used for measuring the EEG signal. The name is derived from 10%-20% distance measures on the head. It considers the distance from the naison (dip between nose and forehead) to the inion (bump at the back of the head above the neck) as 100%. The first line of electrodes (Fp1, F7 . . . O1, O2 . . . F8, Fp2) is placed 10% of this distance up from the naison and inion. Fz is another 20% up, Cz another 20%, etc. (see Figure 2.5.a). For electrode identification, each electrode symbolled with a letter and a number. The letters refer to the lobes of the brain (the Frontal, Temporal, Central, Parietal and Occipital) but there is no central lobe of the brain and this is just an identification term. The even/odd numbers refer to the right/left hemisphere respectively so, the electrode numbers refer to the hemisphere locations. The smaller the number, it is closer to the mid-line between the two hemispheres. The electrodes

on the mid-line represented by a (z), see Figure 2.5.b. Depending on usage, the number of electrodes used can vary (Stephen, 2007).

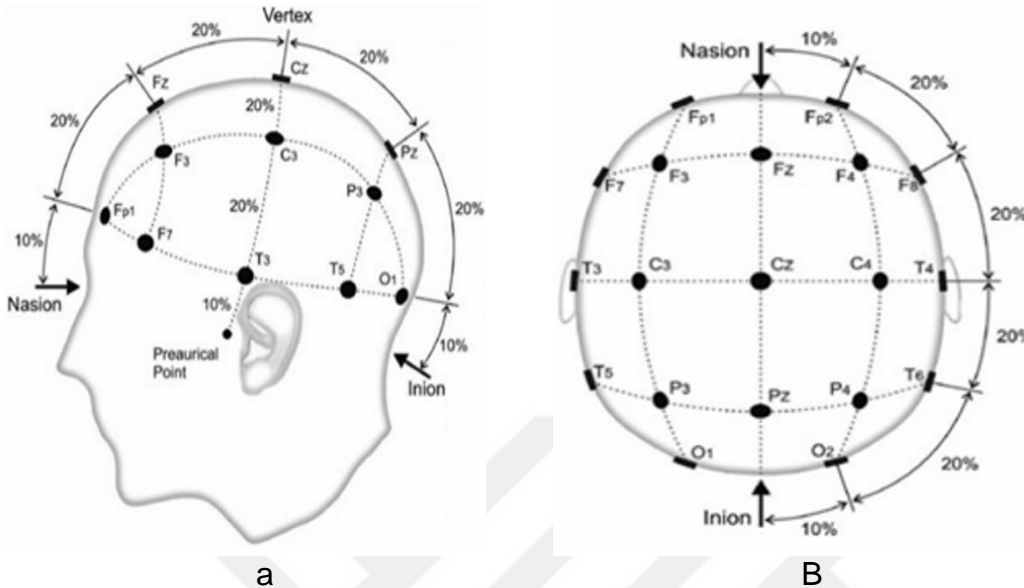


Figure 2.5: The 10-20 System Electrode Placement (Graimann, Allison, & Pfurtscheller, 2010).

2.8 Montages

Each EEG channel is made of two inputs. In EEG recording, one of these inputs is always an electrode and the other input is the reference voltage which will be compared to the first input. There are different approaches for generating this reference voltage. These various configurations are known as montages (Stephen, 2007):

Common Reference: The common reference montage uses a common reference point as one input to each differential amplifier. Each channel of EEG is then formed by the difference between one scalp electrode and a reference electrode. The reference electrode should be chosen as a point which is generally electrically quiet. An advantage to this approach is that the distribution of activity over the scalp is very easy to determine. However, it is often difficult to find a quiet electrode to use as the common reference. The central point Cz is often used as a reference channel to the all electrodes.

Average Reference: This montage is similar to the common reference montage in that, the same reference is used as one input to all the differential amplifiers. However, the common reference is formed by summing all of the activity from the electrodes, averaging it and passing this through a high value resistor. This constructs a quiet

electrode and it also eliminates the problem of the previous montage (Al-Aimama, 2013) (Handy, 2009).

Bipolar: The bipolar montage connects all of the scalp electrodes in a chain form. An electrode which serves as the input for one differential amplifier also serves as the reference for the next. These chains normally run from the front of the head to the back or transversely across the head. The advantage of the bipolar montage is that the activity in neighboring electrodes can be easily distinguished, thereby giving better spatial resolution than the reference montage types. A disadvantage of this approach is that the amplitude and morphology of the activity can be distorted if it affects both electrodes used to generate a channel (Al-Aimama, 2013).

2.9 EMOTIV headset

EMOTIV EPOC+ is a 14 -channel mobile EEG headset and designed for scalable brain research and brain computer interface. It provides easy and quick access to acquire EEG data. It also has a facility to access high-quality raw EEG with a PRO license. It is a wireless rechargeable device with 14 channels based on wet electrodes, see Figure 2.6. The technical specification of this headset are:

- 14 channels: AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF2
- References: In the CMS/DRL noise cancellation configuration P3/P4 locations
- Sampling method: Sequential sampling. Single ADC
- Sampling rate: 128 SPS or 256 SPS* (2048 Hz internal)

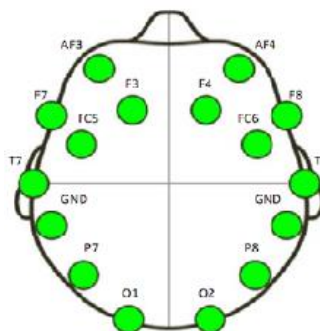


Figure 2.6: Electrodes of EMOTIV EPOC+ Headset.

- Resolution: 14 bits 1 LSB = $0.51\mu\text{V}$ (16 bit ADC, 2 bits instrumental noise floor discarded), or 16 bits*
- Bandwidth: 0.2 – 43Hz, digital notch filters at 50Hz and 60Hz

- Filtering: Built in digital 5th order Sinc filter
- Dynamic range (input referred): $8400\mu\text{V}(\text{pp})$
- Coupling mode: AC coupled
- Wireless: Bluetooth® Smart
- Proprietary wireless: 2.4GHz band
- Battery: Internal Lithium Polymer battery 640mAh
- Battery life: up to 12 hours using proprietary wireless, up to 6 hours using Bluetooth® Smart (EMOTIV, 2008)



CHAPTER 3

DATA ANALYSIS

3.1 Introduction

The aim of data analysis is to extract, to enhance, to compress, or to transmit useful information from the signal. Our environment is filled with various kinds of signals. Some of them are natural and other signals are manmade such as music ..etc. In an engineering context, information is carried through signals and this information may be useful or not. Therefore, signal processing is used to extract or enhance the useful information from a mix of conflicting information (Ingle & Proakis, 2012).

EEG signals are corrupted with different noises or artifacts. Thus the biggest challenge in EEG applications is to reduce the effect of the noise to extract useful features for patterns such as motor movement or a neurological pathology. Such artifacts may be caused by external or internal interferences during EEG recording. The human body physiological activities represent the internal artifacts that interfere EEG such as EMG, ECG and EOG. External artifacts are those who are caused by non physiological activities during the acquisition of EEG signal such as power-line coupling, electrode popping (Maswanganyi, Tu, Owolawi, & Du, 2018) (Guzmán, Heute, Stephani, & Galka, 2017).

Brain computer interface BCI is a system that establishes a communication channel between human and an external device. Such system consists of several parts which are human signal acquisition, preprocessing, feature extraction and signal prediction (Alireza, Esmat, Ali, Mehdi, & Farhad, 2017). Preprocessing stage may involve one or more of numerous methods that try to eliminate the artifacts and to enlarge the signal to noise ratio (Maswanganyi, Tu, Owolawi, & Du, 2018). These methods includes signal filtering, signal processing (Fourier transform, wavelet transform ..etc.), blind source separation, and normalization. Feature extraction try to extract the most discriminative features from EEG signal and try to reduce the signal dimensionality (Muhammad, Nauman, Hubert, & Shum, 2017). Also this stage may involve one or more of statistical methods such as mean, standard deviation ..etc.

This chapter gives brief description about the methods used in this thesis for signal denoising, feature extraction and feature selection. The chapter also involves a description about topic related to our study.

3.2 Wavelet Transform

Perhaps the better way to illustrate wavelet transforms is to first check some transforms and its concepts. The purpose of any transform is to make our job easier. For example, fast Fourier transform (FFT) allows us to see the frequency domain of signals. In other words, it describes the signal in terms of its frequencies by correlating (comparing) the signal with these various sinusoids. Figure 3.1 shows the signal consist from sinusoids of different frequencies (spectrum). Unfortunately, FFT extracts the event frequencies but it losses the time of its occurrence. A possible solution of this problem is to divide the total time into shorter intervals and then implement the FFT for each interval. This provides time-frequency information about the signal and this method called short-time Fourier transform (STFT) (Fugal, 2009).

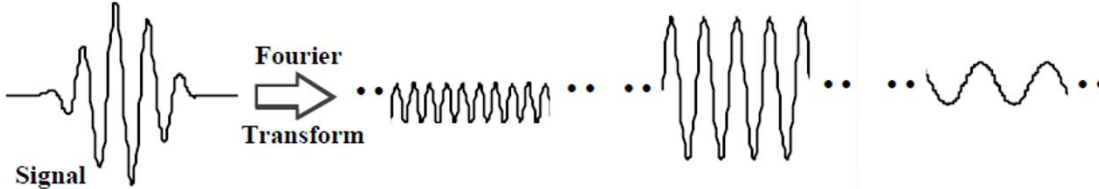


Figure 3.1: Fourier Transform Convert The Signal to Various Sinusoids of Different Magnitudes And Frequencies.

STFT provides time-frequency information about the signal (event) but its accuracy is limited by the size and the shape of the window. For example, a narrow window gives an accurate time resolution but in contrast reflect worse frequency resolution and vice versa.

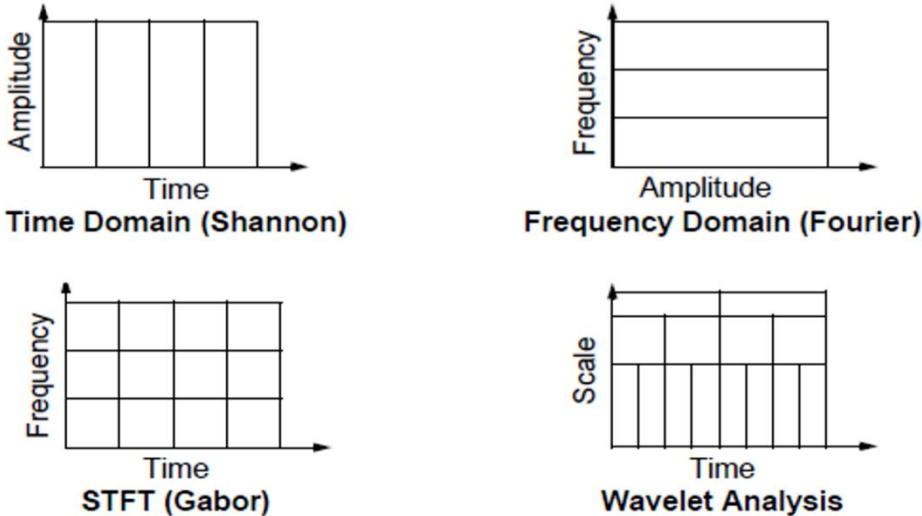


Figure 3.2: The Windowing Comparison Between Three Transformation (FFT, STFT, WT (Misiti, Misiti, Oppenheim, & Poggi (2002)).

Wavelet transform has the facility of variable size windows (Fugal, 2009). Figure 3.2 illustrates the windowing difference between these three transformations (FFT, STFT, and WT).

3.2.1 Continuous Wavelet Transform

The continuous signal can be transformed using a continuous wavelet transform which is defined as

$$T(a, b) = \int_{-\infty}^{\infty} x(t)\psi_{a,b}^*(t)dt \quad (1)$$

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right) \quad (2)$$

Where Ψ is a function called mother wavelet function and a, b are the dilation and location parameter respectively. This function can be stretched or squeezed (dilation) via parameter a or can be moved (translation) via parameter b . Figure 3.3 shows a squeezed and stretched to, respectively, half and double of its original width on the time axis of the one of wavelet function (Mexican hat). The dilation parameter a is the tuning parameter that dilates and contracts the wavelet. The distance between the wavelet crossing of the time axis and its center is represented by this parameter (Addison, 2017).

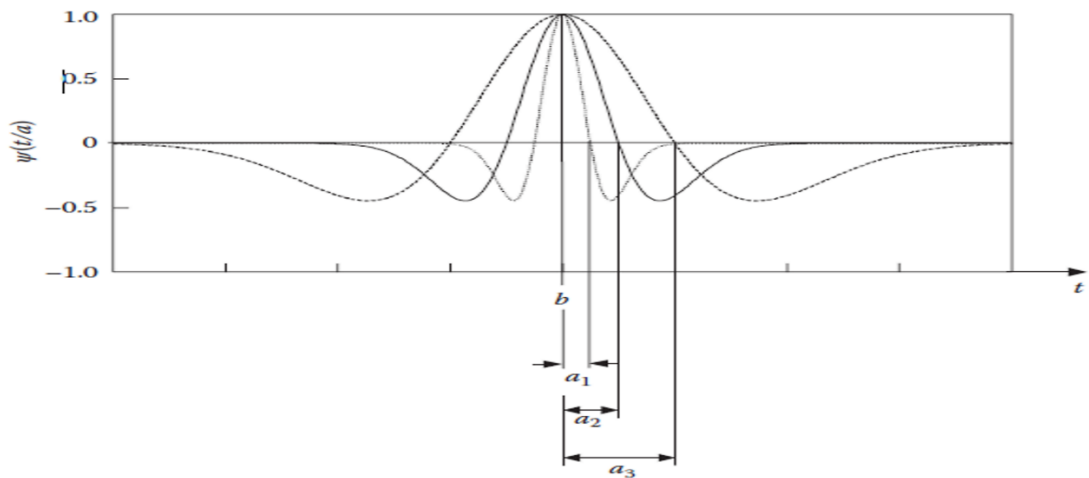


Figure 3.3: Dilation Factor A ($a_1 = a_2/2$; $a_3 = 2a_2$) of The Wavelet Function.

The second parameter b (the translation parameter) moves the wavelet function along the time axis. Figure 3.4b shows the translation of a wavelet from b_1 via b_2 to b_3 along the time axis.

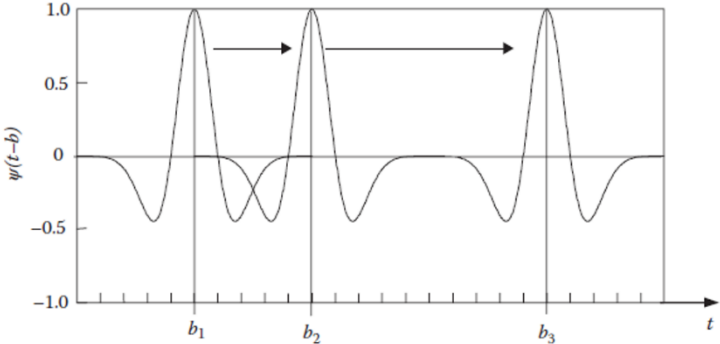


Figure 3.4 : Wavelet Translation.

For instance of WT, let's examine a simple example. Figure 3.5 shows Mexican hat wavelets of various dilations that are correlated with a simple sinusoidal waveform at various locations. The dilation and the location and of the wavelet determines the value of the wavelet transform integral (Equation 1). Let's examine Figure 3.5a, which superimposes the signal with the wavelet at location b and that leads to a local matching between the signal and the wavelet. It can be seen obviously from the figure, the configuration of these two parameters (a and b) makes a high correlation between the signal and wavelet. The integration of the WT ($T(a,b)$) produces here a high positive value caused form the production of the signal with the wavelet. Figure 3.5b shows the wavelet and signal appear to be out of phase after the location of the wavelet was changed to a new location. Here, A large negative value of $T(a,b)$ is produced from the WT integral. The third figure (Figure 3.5c) shows the midpoint between the last two figures. It shows the position of the wavelet with respect to the signal produces a zero value of $T(a,b)$ (Addison, 2017).

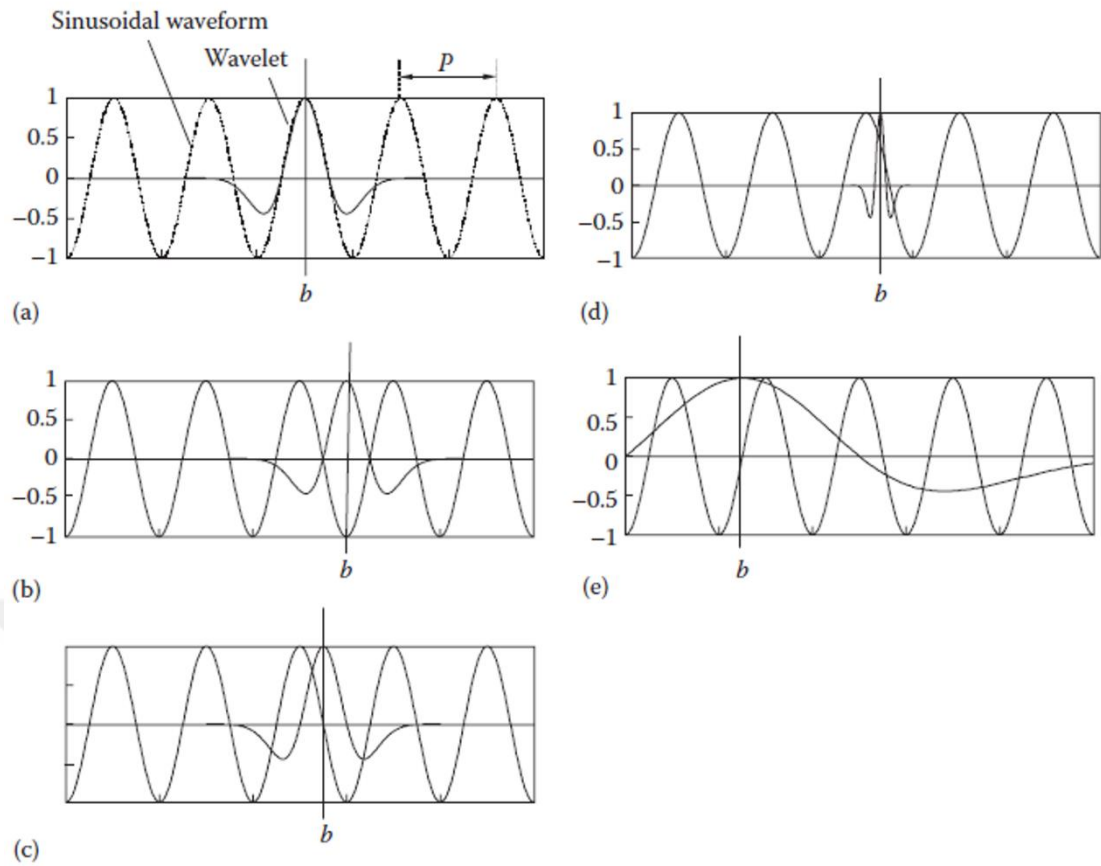


Figure 3.5: Wavelet Transform of Sinusoidal Signal.

The impacts of using a smaller a scale on the transform are shown in Figure 3.5d. It is obviously the negative and positive parts of the wavelet superimposed with almost the same parts of the signal which produces approximately zero value of $T(a,b)$. The same thing occurs in Figure 3.5e when setting a with a large value. That concludes, when the wavelet function approaches to zero width it will make $T(a,b)$ tend to be zero. $T(a,b)$ also tends to be zero when the wavelet covers repetitive negative and positive parts of the signal. Thus, a very small value of the WT means that the wavelet function has either very small or very large width compared with the signal. Usually, this transform uses continuous range of a and b instead of arbitrary dilations and translation parameters (Addison, 2017).

3.2.2 Inverse WT

There is an inverse wavelet transform like Fourier transform and it is defined as

$$x(t) = \frac{1}{C_g} \int_{-\infty}^{\infty} T(a, b) \psi_{a,b}(t) \frac{dad b}{a^2} \quad (3)$$

Where C_g is the admissibility coefficient and represented as

$$c_g = \int_{-\infty}^{\infty} \frac{|\psi(f)|^2}{f} df < \infty \quad (4)$$

3.2.3 Discrete Wavelet Transform

The use of a logarithmic discretization of the two scaling and translating parameters is a natural operation to simplify these two parameters. Also, establishing a link between them makes the WT so easier. To accomplish this, translation factor b is varied in discrete steps and with the proportional to the a scale. The form of the discretization wavelet is illustrated in (Addison, 2017):

$$\psi_{m,n}(t) = \frac{1}{\sqrt{a_0^m}} \psi(a_0^{-m}t - nb_0) \quad (5)$$

where dilation and translation are controlled by integers m and n , a_0 is the dilation step parameter greater than 1 and b_0 is the location parameter greater than zero.

The discrete wavelet transform of a continuous signal becomes in the following form:

$$T_{m,n} = \int_{-\infty}^{\infty} x(t) \frac{1}{a_0^{m/2}} \psi(a_0^{-m}t - nb_0) dt \quad (6)$$

where $T_{m,n}$ are the discrete wavelet transform values given on a scale–location grid of index m, n .

3.2.4 Dyadic Grid Scaling of Wavelet Transforms

Setting a_0 and b_0 to 2 and 1 respectively makes the discrete wavelet parameters having a power of two logarithmic variations which is known as the ‘dyadic grid’ of both the dilation and translation. This arrangement (dyadic grid) is the most efficient and simplest discretization of the wavelet transform. The original signal can be reconstructed using orthonormal wavelet basis ($\psi_{m,n}(t)$) and the wavelet coefficients ($T_{m,n}$) so the *inverse discrete wavelet transform* formula is:

$$x(t) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} T_{m,n} \psi_{m,n}(t) \quad (7)$$

3.2.5 Scaling Function and Multiresolution Representation

There are other orthonormal and dyadic functions associated with wavelet functions called *Scaling function*. The scaling function smooths the signal unlike wavelet function and it has the same form as the wavelet which is given by:

$$\phi_{m,n}(t) = \frac{1}{\sqrt{2^m}} \phi\left(\frac{t - n2^m}{2^m}\right) \quad (8)$$

The convolution of the signal with scaling function produces *approximation coefficients* and its formula is (Addison, 2017)

$$S_{m,n} = \int_{-\infty}^{\infty} x(t) \phi_{m,n}(t) dt \quad (9)$$

The approximation coefficients and the wavelet (detail) coefficients can reconstruct a signal $x(t)$ using the following equation:

$$x(t) = \sum_{n=-\infty}^{\infty} S_{m_0,n} \phi_{m_0,n} + \sum_{m=-\infty}^{m_0} \sum_{n=-\infty}^{\infty} T_{m,n} \psi_{m,n}(t) \quad (10)$$

This equivalent to the following equation

$$x(t) = x_{m_0}(t) + \sum_{m=-\infty}^{m_0} d_m(t) \quad (11)$$

From this equation, it is easy to show that

$$x_{m-1}(t) = \sum_{n=-\infty}^{\infty} S_{m,n} \phi_{m,n}(t) + \sum_{n=-\infty}^{\infty} T_m \psi_{m,n}(t) \quad (12)$$

The last equation tells us that at arbitrary scale (m), we can add both the signal approximation and detail coefficients to the signal approximation at higher resolution

(i.e. at a smaller scale $m - 1$). This is called a *multiresolution representation* (Addison, 2017).

3.2.6 Fast Wavelet Transform FWT

At the end of the 1980s, Stéphane Mallat proposed a fast algorithm of decomposition-reconstruction for the discrete wavelet transform. He thus established the link between the traditional filter banks in signal processing and the orthonormal wavelet bases. This algorithm is remarkably simple and it has linear complexity with the size of data which means, it is lower than the fast Fourier transform. This aspect is obviously crucial for applications (Misiti, Misiti, Oppenheim, & Poggi, 2007).

This algorithm is a fast algorithm for machine computation, like the Fast Fourier Transform (FFT). It transforms the vector into a numerically different vector of the same length. Also, like the FFT, the DWT is invertible and orthonormal. DWT has a hierarchical set of “wavelet functions” that satisfy certain mathematical criteria (Daubechies, 1992; Mallat, 1989b) and are all translations and scaling of each other. Unlike FFT, the base functions are sines and cosines.

3.2.6.1 Fast wavelet decomposition

Fast wavelet transform (FWT) is a multiresolution analysis that uses a pair of quadrature mirror filters defined from the underlying wavelet function. This algorithm is the equivalent version of DWT but faster than it. In this process, the signal is split into two components, the first contains the low-frequencies or “approximation coefficients” information and the other contains the high-frequencies or “detail coefficients” information. The two components have half of the original signal length (Olkkonen, 2011). These two coefficients can be calculated using equations 13 & 14:

$$C_j(k) = \sum_m h_d(m - 2k) c_{j+1}(m) \quad (13)$$

$$d_j(k) = \sum_m g_d(m - 2k) c_{j+1}(m) \quad (14)$$

where c and d are the scaling and details coefficients. These two coefficients can be computed according to equations 13 & 14 at different levels of scale by convolving the expansion coefficients at scale $j+1$ by low and high pass filters ($h_d(n)$ and $g_d(n)$ respectively) then down-sampling by 2 to give the scaling (approximation) and details coefficients at the next level of j . Figure 3.6 illustrates the function of equations 13 and 14 (FWT) (Burrus, Gopinath, & Guo, 1998) (Fugal, 2009).

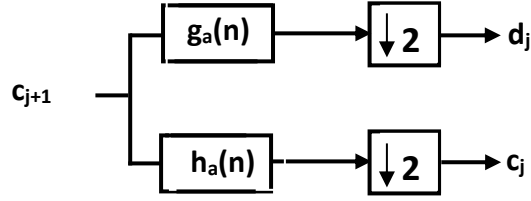


Figure 3.6: Single Level Fast Wavelet Transform (Burrus, Gopinath, & Guo, 1998).

This procedure can be repetitively repeated on the scaling coefficients and this method is called multiresolution FWT (Burrus, Gopinath, & Guo, 1998) (Fugal, 2009). Figure 3.7 shows the two decomposition stages of FWT. This multiresolution decomposition of a signal into its scale and detail components is useful for data compression, feature extraction, and denoising (Rao & Yip, 2001)

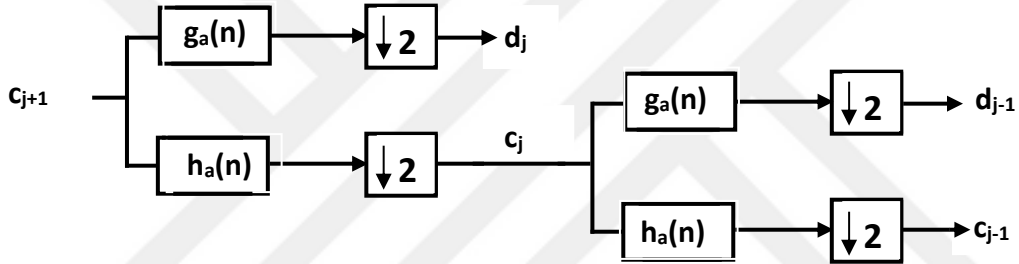


Figure 3.7: Two Stage Decomposition of FWT (Burrus, Gopinath, & Guo, 1998).

3.2.6.2 Fast wavelet reconstruction

The original signal can be reconstructed from a combination of the scaling and details coefficients at a coarse resolution. Equation 15 represents the synthesis equation of FWT. In signal synthesis, the coarse two coefficients (scale and details coefficients) are first upsampled by inserting zeros between its coefficients and then filtered by two reconstruction filters ($h_s(n)$ and $g_s(n)$ which are low and high pass filters respectively). This operation also can be repetitively repeated from coarse level ($j-1$) to fine level j many times to perform multiresolution FW synthesis. Figure 3.8 shows two stages of FW synthesis (Burrus, Gopinath, & Guo, 1998) (Fugal, 2009).

$$C_{j+1}(k) = \sum_m C_j(m)h_s(k - 2m) + \sum_m d_j(m)g_s(k - 2m) \quad (15)$$

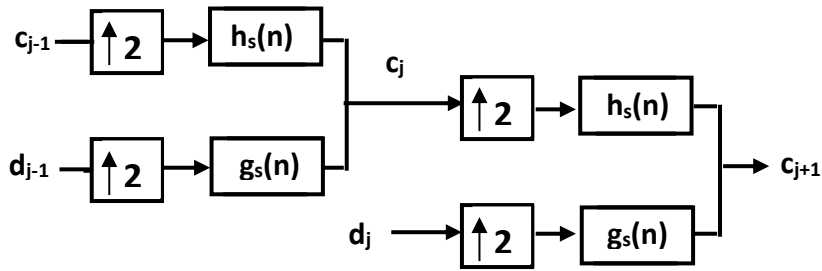


Figure 3.8: Two Stages of FW Synthesis (Burrus, Gopinath, & Guo, 1998).

3.2.7 Thresholding and Denoising

There are a lot of de-noising methods, such as time domain analysis, spectrum analysis, adaptive filtering, wavelet transform de-noising. Among them, the de-noising method based on wavelet thresholding has a good treatment effect. This method has a simple and small amount of calculation and it is applied more widely. Hard-threshold and soft-threshold method are the common de-noising methods based on wavelet threshold. These two methods are magnitude thresholding which is normally carried out to smooth the data, remove noise from a signal or to partition signals into two or more (and not necessarily noisy) components. They separate out the behavior of interest from the signal through the reduction or complete removal of the selected wavelet coefficients in order. Their behavior is shown schematically in Figure 3.9 (Addison, 2017) (Golilarz & Demirel, 2017).

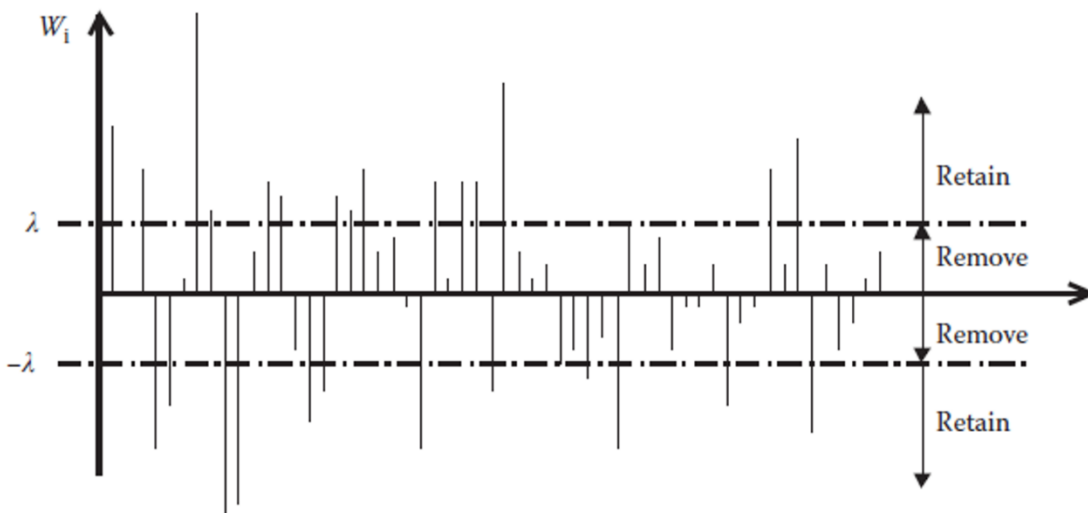


Figure 3.9: Schematical Behavior of Magnitude Thresholding.

In hard thresholding, wavelet coefficients will be kept if their absolute values are larger than the threshold value, otherwise, they will be set to zero. In soft thresholding, all wavelet coefficients will be shrunk by the threshold value, if their absolute values are

larger than threshold value otherwise the coefficients will be zero. Hard and soft thresholding equations are given by (16) and (17), respectively (Yang, Deng, Chen, & Xu, 2014).

$$W_i^{hard} = \begin{cases} 0 & |W_i| < \lambda \\ W_i & |W_i| \geq \lambda \end{cases} \quad (16)$$

$$W_i^{soft} = \begin{cases} 0 & |W_i| < \lambda \\ \text{sign}(W_i)(|W_i| - T) & |W_i| \geq \lambda \end{cases} \quad (17)$$

where λ is the threshold value and W are the wavelet coefficients. Figure 3.10 shows both hard and soft thresholding schematic diagram relationship between the original and thresholded coefficients (Addison, 2017). Nowadays, choosing a proper threshold value to remove noisy coefficients from the signal to produce a clean version of the signal is a crucial demand for many researchers (Golilarz & Demirel, 2017).

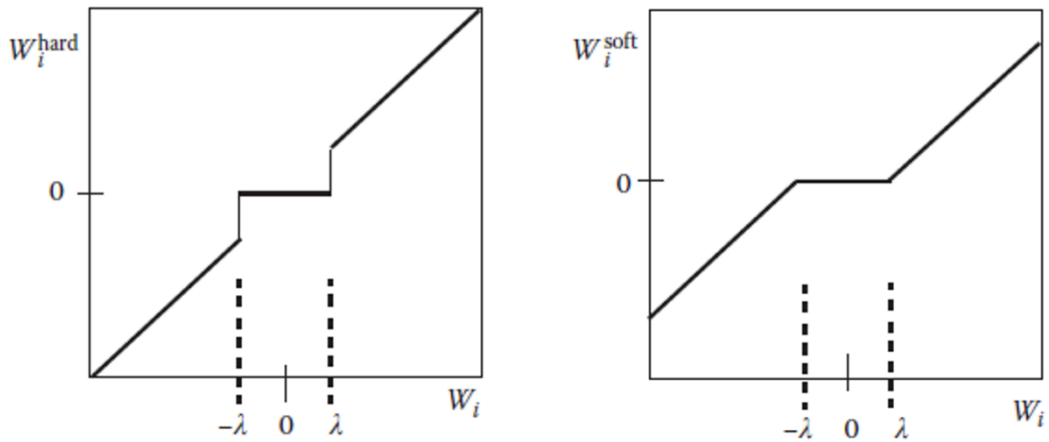


Figure 3.10: Relationship Between The Original Wavelet Coefficient And The Thresholded Coefficients.

3.3 Feature Extraction

Feature extraction techniques usually convert signal or data to reduce its dimension and to extract useful information from the data (Muhammad, Nauman, Hubert, & Shum, 2017). Data can be characterized according to its central tendency and dispersion. *mean*, *median*, *mode*, and *midrange* are the methods that are used to measure data central tendency while *quartiles*, *interquartile range (IQR)*, and *variance*

are used to measure data dispersion. These statistical methods provide a helpful understanding of data distribution (Jiawei & Micheline, 2006).

3.3.1 Correlation

Correlation is a tool to measure the similarity degree between two signals. It has enormous applications such as sonar, radar, geology ..etc. Let $x(n)$ and $y(n)$ be two different signals so cross-correlation (r_{xy}) of these two signal can be calculated as:

$$r_{xy}[m] = \sum_{-\infty}^{\infty} x[n]y[n - m] \quad m = 0, \pm 1, \pm 2, \dots \quad (18)$$

where m is the time shift (lag) parameter and the subscripts xy of cross-correlation indicates the sequences being correlated. That's in equation 18, only the signal $y(n)$ is shifted from right to left with respect to $x(n)$ signal when m changes from positive to negative. When $x(n)=y(n)$ which is a special case, the operation is called autocorrelation. Autocorrelation is defined as (Proakis, 2006):

$$r_{xx}[m] = \sum_{-\infty}^{\infty} x[n]x[n - m] \quad m = 0, \pm 1, \pm 2, \dots \quad (19)$$

In the case of a finite sequence (M) of $x(n)$ and $y(n)$, the cross-correlation is expressed as:

$$r_{xy}[m] = \sum_{i=0}^{N-|m|-1} x[i]y[i - m] \quad (20)$$

where $m = [-(N - 1), -(N - 2), \dots, 0, 1, 2 \dots (N - 1), (N - 2)]$ and $r_{xy}[m]$ has $2M-1$ samples, if $x[n]$ and $y[n]$ have equal number of samples (M) (Al- Dabag & Ozkurt, 24 October 2018) (Proakis, 2006) .

3.3.2 Measuring the Central Tendency

Average or *arithmetic mean* or simply *mean* is a numerical tool to measure the data center. Its formula is

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad (21)$$

where x_i are the N data observations. Although, this method not always the best method for measuring the center of data because of its sensitivity to the outliers (extreme values).

Median is another and better tool to measure the center of data. To evaluate it, the data has to be sorted first. Then, if the data length is odd, the middle value of the sorted data is its median. Otherwise (i.e., if the data length is even), the average of the two middle values represents the data median.

Mode also another method to evaluate the central tendency of the data. The mode value is a value that occurs most frequently in the data set. It is possible to have more than one mode value in a dataset. On the other hand, there may be no mode in the dataset, if each element occurs only one (Jiawei & Micheline, 2006).

3.3.3 Measuring the Dispersion of Data

The range of a set is defined as the difference between the largest and smallest values. The center, shape, and spread of data distribution can somewhat be indicated by the quartiles, including the median. The first quartile (Q1) is equivalent to the median of the first half of the sorted data while the third quartile (Q3) represents the median of the second half of the sorted data. The middle point between Q1 and Q2 is called interquartile range (*IQR*). The shape distribution of the data set can be summarized with only five numbers which are the smallest, Q1, median, Q3, and the largest values of the data set (Jiawei & Micheline, 2006).

Variance and Standard Deviation are other tools to evaluate the data dispersion. The variance of the data set with N observations can be calculated by

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2 \quad (22)$$

The standard deviation σ is the square root of the variance σ^2 . Standard deviation σ qualifies how much the members of a dataset differ from the mean value for that dataset (Jiawei & Micheline, 2006).

3.4 Feature Selection

A feature selection or reduction process can be used to eliminate the redundant terms in the dataset. This will reduce the set of terms to be used in classification, thus improving both efficiency and accuracy (Jiawei & Micheline, 2006). A smaller amount of features is always preferable because it yields: higher speed, higher mobility, and

lower the costs of implementing the BCI, and above all—higher comfort for its user (Izabela, 2014). Two models of feature selection exist depending on whether the selection is coupled with a learning scheme or not. The first one, the filter model, which carries out the feature subset selection and the learning (e.g., classification, clustering) in two separate phases, uses a measure that is simple and fast to compute. The second one, the wrapper method, which engages the feature selection with the learning algorithms in the same process to measure the accuracy of the extracted features (Talbi, 2009).

Optimized, randomized, and heuristical search algorithm are three different methods that can be used for searching the feature subset. Randomized algorithms have two significant properties or at least one of them which are the fastest and simplest algorithms. These two properties make the randomized search algorithm suitable for many applications. These algorithms provide a magic solution for searching into huge feature search space. Unlike deterministic search algorithms, it is often used to avoid falling into local optima (Huan & Hiroshi, 2008). For these reasons, we have used the genetic algorithm for feature selection. In the next section, genetic algorithms will be summarized briefly.

3.4.1 Genetic Algorithms (GA)

Genetic algorithms are a general purpose mechanism for randomized search. There are four key aspects to their use: encoding, population, operators, and fitness. Figure 3.11 summarizes the genetic algorithm into four steps. The individual states in the search space (chromosome) must be first encoded into some string-based format, typically bit-strings (Huan & Hiroshi, 2008). The major steps of GA are illustrated in the following steps:

Step 1: initiate: chromosome length which represents the problem variable domain, the population size (N), the mutation probability p_m , and the crossover probability p_c .

Step 2: To evaluate each individual performance, fitness function has to be defined. The GA depends on this function for selecting the parent chromosomes during reproduction.

Step 3: Random initialization of N chromosomes population:

$$x_1, x_2, \dots, x_N$$

Step 4: evaluate each individual chromosome using the fitness function.

Step 5: from the current population, mating parent chromosomes are selected proportionally with their fitness cost. So, the lowest fitted chromosome has the lowest chance to be selected for mating and vice versa.

Step 6: apply two GA operators (crossover and mutation) on the selected parent pairs to create a new offspring chromosome pairs.

Step 7: reserve the created offspring chromosomes in the new population.

Step 8: Repeat Step 5 until the size of the new population is equal to population size (N).

Step 9: Replace the new offspring population instead of the parent chromosome population.

Step 10: Go to Step 4, and repeat the process until the termination criterion is satisfied.

Since, the population fitness may remain stable during the search progressing so the common solution is to set the termination condition with specific number of generations. If the GA solution does not satisfy some criteria, the GA could be restarted again (Michael, 2005).

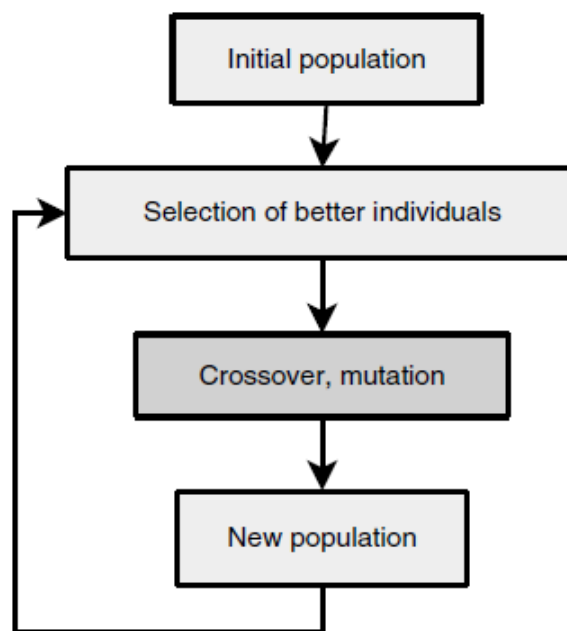


Figure 3.11: Steps of Genetic Algorithm (Talbi, 2009).

3.4.1.1 Selection Methods

The selection methods have a main principle which is to choose the better individual that has higher fitness. Such a selection will lead to drive better solutions. However,

this doesn't mean that the worst individuals has no chance to be selected. The most popular selection methods are roulette wheel selection and tournament selection.

In Roulette wheel selection, suppose each individual has a space in a pie graph proportional to its fitness, see Figure 3.12. The outer roulette wheel will spin N times (the size of the population) to select N individuals. Each individual with a large space in the pie has more opportunity to be chosen.

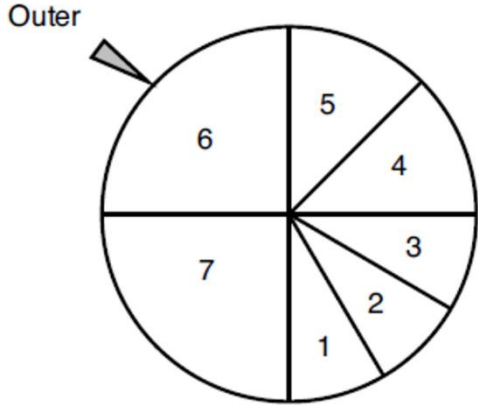


Figure 3.12 : Roulette Wheel Selection (Talbi, 2009).

In tournament selection, a specific size of tournament group (k) is to be set by the user. First, K individuals are selected randomly and then the best individual within this group will be chosen. Tournament selection repeats this procedure N times to generate a complete population. Figure 3.13 illustrates this selection method (Talbi, 2009).

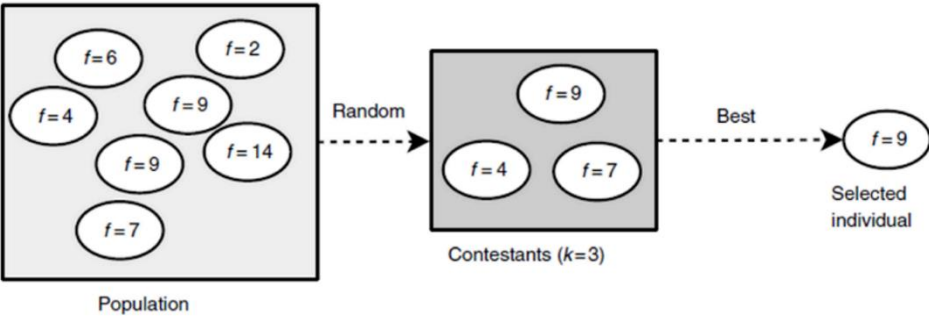


Figure 3.13: Tournament Selection (Talbi, 2009).

3.4.1.2 Crossover

Crossover operator tries to create a new individual that are better than his parents. This is done if it inherits the best features from each of them. The occurrence probability of this operator is defined by the user. The crossover has different strategies to operate which includes:

- One point: a one crossover point is randomly selected within a chromosome then the two parent chromosomes interchange their segments at this point to produce two new offspring. See Figure 3.14.
- Two points: a similar procedure is done, like one point crossover, but two crossover points are randomly selected.
- Uniform: the interchanging occurs randomly at the gene level rather than at the segment level to mix the parent chromosomes and it is also done with some probability to generate the new offspring (Silva, Bellon, & Boyer, 2005).

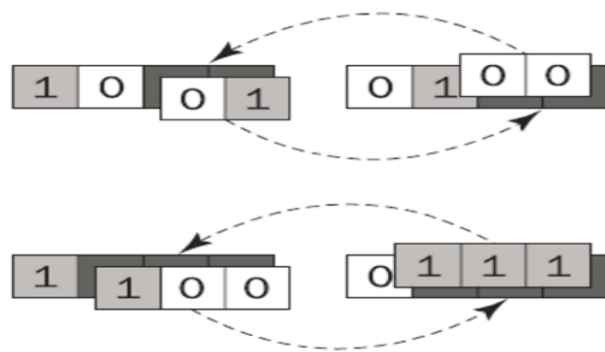


Figure 3.14: One Point Crossover (Michael, 2005).

3.4.1.3 Mutation

Mutation operator toggles a randomly selected bit in an individual string with a low probability. Figure 3.15 illustrates the mutation operation. This operator prevents the algorithm from trapping into a local optimum. It may lead the algorithm to significant improvement on its search, but more often mutation operator has harmful effects so it applied at a low probability (Michael, 2005).

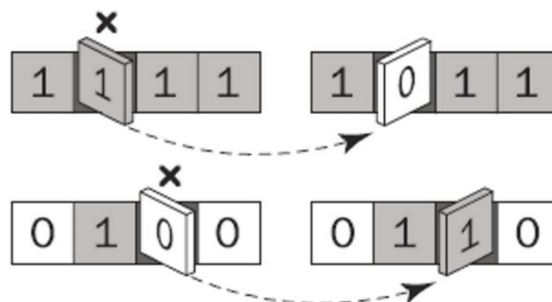


Figure 3.15 : Mutation Operation (Michael, 2005).

CHAPTER 4

MACHINE LEARNING AND PARALLEL PROCESSING

4.1 Introduction

Scientists try to understand the way that human beings think and make machines act like humans. Many algorithms has been proposed to simulate human intelligence to make machines solve problems and make decisions. The crucial demands for many machine learning applications are accuracy and speed. Usually, these algorithms have huge complexity so it is hard to be executed using an ordinary machine (sequential computers). Parallel processing and programming provides a suitable solution for these kinds of algorithms.

This chapter consists of two parts. The first one describes the most popular two machine learning algorithms which are the neural network (multilayers perceptron networks with backpropagation learning) and support vector machines as well as its evaluation methods. The second part summarizes one of the most famous parallel processing technique which is machine pipelining. It also gives a brief description of concurrent processing techniques.

4.2 Machine Learning Algorithms for Classification

This section will provide a brief description of the evaluation methods to evaluate the performance of a classifier. It also describes the proper situation suited for each evaluation method. After that, it will provide a brief description of the most two popular classifiers used in this thesis which are multilayer perceptron (MLP) and Support vector machine (SVM).

4.2.1 Measuring the Classifier Accuracy

The classifier performance evaluation is based on the counting the test tuples that correctly and incorrectly predicted by the classifier. The confusion matrix summarizes this evaluation. Table 4.1 shows the binary classification problem confusion matrix. The positive class is normally the utmost class to the designer and the true positive

(TP) is the number of correctly predicted data tuples as a positive class while the false negative (FN) is the number of the wrongly predicted data tuples as a positive class. True negative (TN) represents the number of the data tuples that are correctly predicted as negative class while the false negative (FN) depicts the number of the incorrect data tuples predicted as negative class. Although, the confusion matrix gives detail information about the classifier performance but summarizing this information with a single number would make it easy and convenient for comparing it with the other model performance. There are more than one method to perform this job. Accuracy is one of these methods which is the percentage of the data tuples that are predicted correctly (Tan, Streinbach, & Vipin, 2005):

$$Accuracy = \frac{\text{number of correct predictions}}{\text{total number of predictions}} = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

Table 4.1: A Confusion Matrix for A Binary Classification Problem.

		Predicted class	
		Positive class	Negative class
Actual class	Positive class	TP	FN
	Negative class	FP	TN

The sensitivity and specificity measurement are other methods to summarize the confusion matrix. *Sensitivity* (also called *recall rate [RR]* or *true positive rate [TPR]*) is the ratio of positive tuples that are predicted correctly, see equation 2.

$$Sensitivity = \frac{\text{number of correct prediction as positive class}}{\text{total number of positive class}} = \frac{TP}{TP + FN} \quad (2)$$

Specificity (*true negative rate [TNR]*) is the ratio of negative tuples that are predicted correctly, see equation 3 (Awad & Khanna, 2015) (Tan, Streinbach, & Vipin, 2005).

$$Specificity = \frac{\text{number of correct prediction as negative class}}{\text{total number of negative class}} = \frac{TN}{TN + FP} \quad (3)$$

4.2.2 Evaluating the Performance of a Classifier

There are some techniques to verify classifiers performance. These techniques randomly sample a given data into partitions for assessment a classifier accuracy, e.g. holdout, random subsampling, cross-validation, and the bootstrap. It increases the

overall computation time but using such techniques are useful to select an effective model.

4.2.2.1 Holdout Method and Random Subsampling

The holdout method splits the data randomly into two sets (training set and testing set). Typically, a given data is sampled randomly and partitioned into three partitions. Training set has two-thirds of the partitioned data and the other one-third is allocated for the testing set. Training the classifier is done by the training set while evaluating its performance is done by the testing set. Figure 4.1 illustrates the holdout estimation method. Random subsampling is similar to holdout method except it repeats running the method k times. The overall classifier accuracy is estimated by taking the average of the k times accuracies (Jiawei & Micheline, 2006).

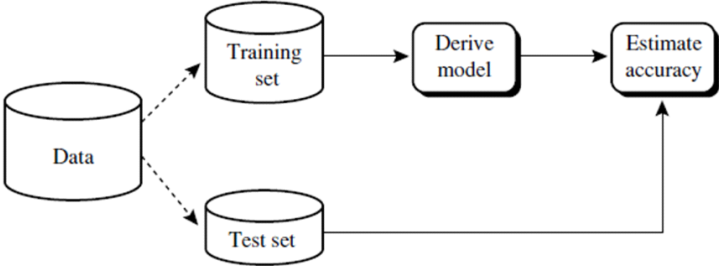


Figure 4.1: Holdout Method for Estimating Classifier Accuracy (Jiawei & Micheline, 2006).

4.2.2.2 Cross-validation

This method randomly samples the data into k subset (folds) in which, each element in a fold is not included in other folds. Each fold has approximately equal size. The classifier estimation is performed k times. In each estimation, one of the fold is used for testing and the others are used for training. The model accuracy estimation is equal to the average of the k iterations accuracies. This procedure (cross-validation) is performed whenever the available dataset is limited.

Leave-one-out cross-validation is equivalent to k-fold cross-validation but the k is the number of instances (elements) in the dataset. The estimation of the model performance is performed in a similar way as k-fold cross validation. leave-one-out is used to estimate a classifier performance whenever the dataset is so limited (Jiawei & Micheline, 2006).

4.2.3 Multilayer Neural Networks

Multilayer neural network is also called multilayer perceptron. It is a feedforward network so, the data flows in one direction from the input layer to the output layer. Typically, the network is constructed from two computational layers and the input layer. The computational layers are the output layer and at least one hidden layer. Figure 4.2 shows a multilayer perceptron network (Michael, 2005).

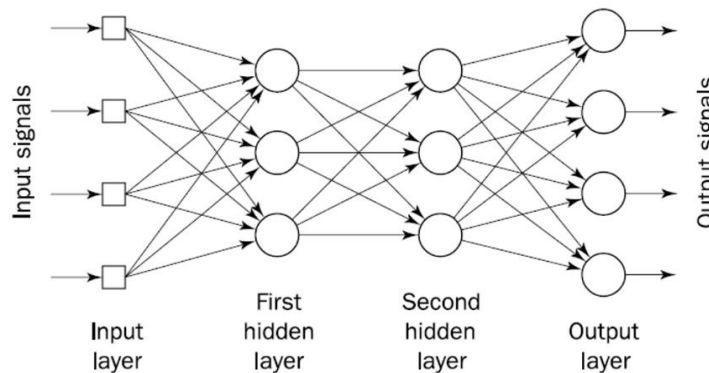


Figure 4.2: Multilayer Perceptron with Hidden Layers.

Learning these networks needs two aspects to be known which are the learning algorithm and the structure of the network. The learning algorithm updates the connection weights of the computation layers. There is a relatively simple learning algorithm for a predefined neural network. This algorithm is known as backpropagation. The following section describes this algorithm.

4.2.3.1 Backpropagation algorithm

The most popular learning algorithm among hundreds of different algorithms is the backpropagation algorithm. This algorithm has two phases. The first, the neural network input layer is supplied by a training set. The multilayer perceptron forwards the training set from layer to layer until reaching the output layer to generate the output pattern. In the second phase, the output pattern is compared with the desired output to calculate the output error. This error (if exists) is then backwarded from the output layer to the input layer. Through error backwarding, the weights of the network are updated. This procedure is repeated until the error is minimized to a specific range. Figure 4.3 illustrates the backpropagation learning algorithm (Michael, 2005).

To illustrate the back-propagation learning algorithm, consider the neural network consists of three-layer as shown in Figure 4.3. The neuron indices of the input, hidden and output layers are i, j, k respectively so the back-propagation training algorithm is:

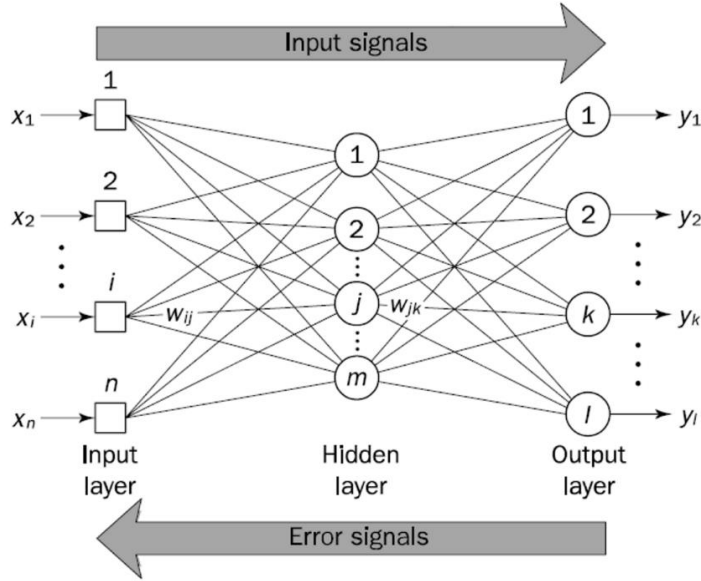


Figure 4.3: Three layer backpropagation neural network.

Step 1: At first iteration $p = 1$, weights are initialized to small random values.

Step 2: For every training pair (training input x_i , desired output y_{di} , where $i=1, \dots, n$), do step 3-4

Step 3: Forward the input data from the input layer to output layer.

a) The outputs of the neurons in the hidden layer are evaluated as:

$$y_j(p) = \text{sigmoid} \left[\sum_{i=1}^n x_i(p) * w_{ij}(p) - \theta_j \right]$$

where sigmoid is the sigmoid activation function and n is the number of the input of the j th neuron in the hidden layer.

(b) The outputs of the neurons in the output layer are also evaluated as:

$$y_k(p) = \text{sigmoid} \left[\sum_{j=1}^m x_{jk}(p) * w_{jk}(p) - \theta_k \right]$$

where m is the number of inputs of k th output neuron in the output layer.

Step 4: update the network Weights (backward error)

(a) For each neuron in the output layer, evaluate its error as:

$$\delta_k(p) = y_k(p) * [1 - y_k(p)] * e_k(p)$$

where

$$e_k(p) = yd_k(p) - y_k(p)$$

The evaluation of the weight corrections is done as:

$$\Delta w_{jk}(p) = \alpha * y_j(p) * \delta_k(p)$$

The weights of the output neurons are updated as:

$$w_{jk}(p + 1) = w_{jk}(p) + \Delta w_{jk}(p)$$

(b) For each neuron in the hidden layer, evaluate its error:

$$\delta_j(p) = y_j(p) * [1 - y_j(p)] * \sum_{k=1}^i \delta_k(p) w_{jk}(p)$$

The evaluation of the weight corrections is done as:

$$\Delta w_{ij}(p) = \alpha * x_j(p) * \delta_j(p)$$

The weights of the hidden neurons are updated as:

$$w_{ij}(p + 1) = w_{ij}(p) + \Delta w_{ij}(p)$$

Step 5: Increase iteration p by one, until the specified error is satisfied go back to Step 2 (Michael, 2005) (Fausett, 1994).

This algorithm modifies the weights of the network based on one of the optimization methods. The simplest optimization method is the *gradient descent* method. The *Newton* algorithm and *Levenberg-Marquardt* algorithm are other optimization methods which are developed based on the gradient descent method (Simon Haykin, 2009). Unfortunately, this algorithm converges slowly and lacks optimality (Simon Haykin, 2009).

4.2.4 Support Vector Machines

The SVM is a supervised learning classifier. It is formulated to classify a nonlinear binary problem. The main objective of SVM is to construct an optimal separating plane (hyperplane) that has a maximum separation margin between the two classes. To understand the basic concepts of SVM, let's consider to classify a simple linearly separable training set using SVM. This data set is formed as

$$\{(x_i, d_i)\}_{i=1}^N \tag{4}$$

where x_i are N -instances vector and d_i are its corresponding target output and d_i are represented as

$$d_i = \begin{cases} +1 & \text{if } x_i \text{ in class 1} \\ -1 & \text{if } x_i \text{ in class 2} \end{cases} \quad (5)$$

There are infinite hyperplanes for linearly separating these two classes but only one of them represents the optimal hyperplane. This hyperplane has distance as maximum as possible between itself and the closest points of the two classes. The optimal hyperplane can be represented as:

$$wx + b = 0 \quad (6)$$

where w is the tuning weight vector, b is the bias and x is the input vector. This equation represents the decision surface and can be found such that it has $wx+b \geq +1$ for positive class and it has $wx+b \leq -1$ for the negative class. The combination of these two equations is:

$$d_i(wx + b) - 1 \geq 0 \quad (7)$$

The distance from the optimal hyperplane to the origin is $(\|b\|)/(\|w\|)$, where $\|w\|$ is the Euclidean norm of w . Figure 4.4 shows the optimal hyperplane with its support vectors (Chandaka, Chatterjee, & Munshi, 2009).

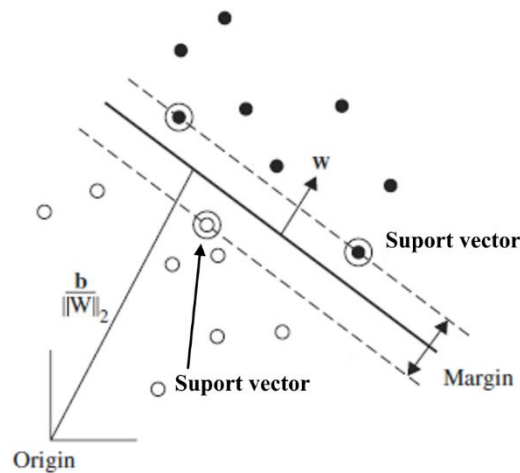


Figure 4.4: The SVM for The Linearly Separable Case.

The data points that, satisfy $wx+b=\pm 1$, constitute the support vectors. These points are the closest points to the optimal hyperplane and represent the most difficult points to be classified. To accomplish this, the optimal hyperplane parameters (w_0 and b_0) have to satisfy a maximum margin of separation for a given training samples (eq. 4). $2/(\|w\|)$ is the distance between the two support vectors (Chandaka, Chatterjee, & Munshi, 2009). SVM target is to maximize this distance by the following optimization problem

$$\text{minimize } \left\{ \frac{1}{2} \|w\|^2 \right\} \quad (8)$$

Under the constraints of (7)

However, in many empirical situations, the datasets are not linearly separable. Therefore, the constraint (7) need to be modified to make the algorithm more flexible. So, all points that fall on the wrong margin are considered to be an error points. Hence, a new set of variables is added to the optimization problem and these variables are called slack variable (see Figure 4.5).

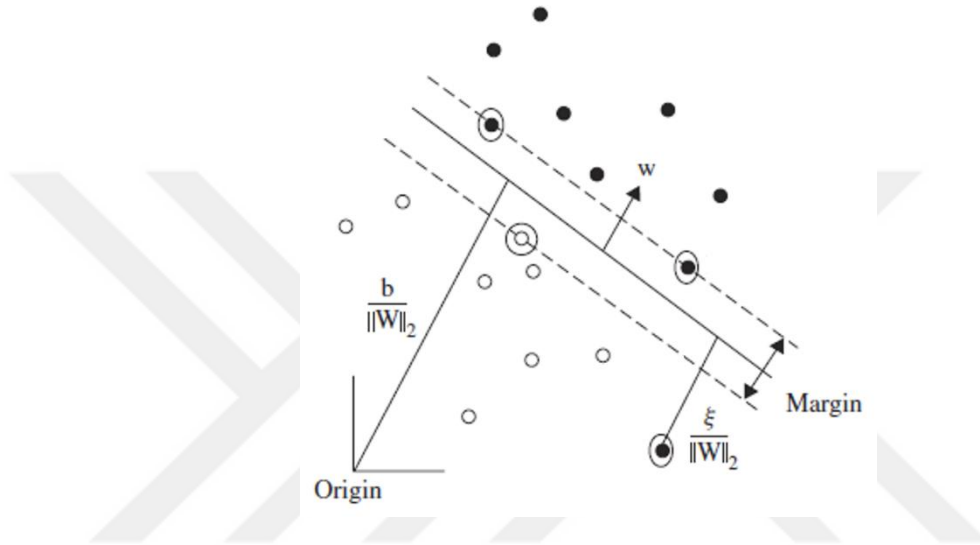


Figure 4.5: The Concept of The Slack Parameter.

The modified optimization problem becomes:

$$\text{minimize } \left\{ \frac{1}{2} \|w\|^2 \right\} + C \sum_{i=1}^N \xi_i \quad (9)$$

$$d_i(wx + b) \geq 1 - \xi_i \quad \text{and } \xi_i \geq 0 \quad i = 1, \dots, N \quad (10)$$

The $\sum_i \xi$ is the maximum error bound of the training errors (Sanei & Chambers, 2007). The variable C tunes the error bound for the misclassified training samples. A small value of C may lead an inappropriate large fraction of support vectors while a large value of C may cause overfitting (Foody & Mathur, 2004).

An alternative solution for non-separable dataset is by using a kernel function. These function project the training data into higher-dimensional space to separate them linearly (i.e. linear hyperplane separator). This converts a simple linear classifier into a powerful nonlinear classifier by using kernel functions. Some of the popular SVM kernel functions are; Polynomial, Exponential radial basis function, Multilayer perceptron, and Gaussian radial basis function (Sanei & Chambers, 2007).

4.3 Basic Pipeline Concepts

Pipelining is an implementation technique to improve process execution rate by overlapping its execution with other processes. It makes use of parallelism that exists among the actions needed to execute a process. Normally, every process can be accomplished through the execution of different distinct steps. In a non pipeline system, these steps must be accomplished before the next process can be issued while in a pipelined system, overlapped fashion is used to execute successive processes, see Figure 4.6.

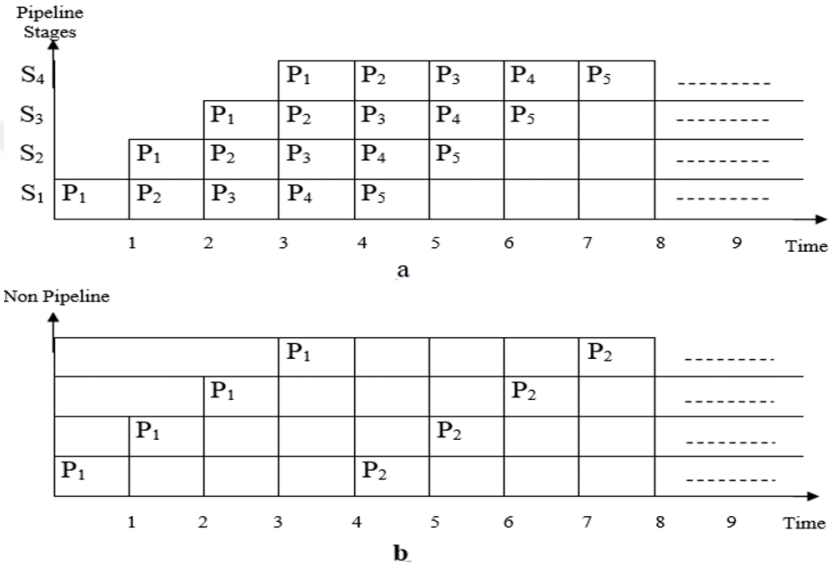


Figure 4.6: Pipeline Concepts (a) Pipeline Timing Diagram (b) Non Pipeline Timing Diagram.

To construct a pipeline, a process has to be segmented into a sequence of subtasks. Each subtask can be processed by a specialized processor and these processors (stages) operate concurrently with others in the pipe. This structure (pipeline) gives facility to process successive tasks in an overlapped fashion (Kai & Faye, 1984). It simulates an assembly line. There are many steps in an automobile assembly line, each contributing something to the construction of the car. Each step operates in parallel with the other steps but on a different car. In a computer pipeline, each step in the pipeline completes a part of a process. Like the assembly line, different parts of different processes are manipulated in parallel on each steps of the pipe. Each of these steps is called a *pipe stage* or a *pipe segment*. The stages are connected sequentially to form a pipe.

The throughput of an automobile assembly line is defined as the number of complete cars exit the assembly line per hour. Likewise, the process pipeline throughput is

measured by how often a process exits the pipeline. All the pipe stages must proceed synchronously because the pipe stages are hooked together, just like in an assembly line so processor cycle is the time required to move a process one step through the pipeline. For this reason, the slowest pipe stage determines the processor cycle, similar to an auto assembly line the advancing time in the line is restricted to the longest step. Balancing the processing time of pipeline stages is the goal of the pipeline designer. If the stages are perfectly balanced, then the time per process on the pipelined processor is measured by

$$\text{pipeline throughput} = \frac{\text{Time per process on unpipelined machine}}{\text{Number of pipe stages}} \quad (11)$$

The number of pipe stages determines the amount of speedup from pipelining for ideal conditions, just as an assembly line with n stages can ideally manufacture cars n times faster than nonpipeline system. However, perfectly balanced stages do not exist; furthermore, some overheads are involved while implementing the pipeline. Thus, the time per process on the pipelined processor can be close but it will not have its minimum possible value (John & David, 2012).

4.4 Concurrency and semaphore

The simple basic principle of the semaphore is two or more processes can cooperate with each other using simple signals by which a process can be forced to stop at a specified condition until it has received an appropriate signal. Suitable structure of signals can satisfy any complex coordination requirement. Special variables called semaphores are used for signaling (William, 2012).

4.4.1 The Producer/Consumer Problem

In the computer science viewpoint, a classical example of the multi-process synchronization problem is the producer-consumer problem. Designing and developing of the software has an important and familiar problem which is the Synchronization problem. Synchronization is needed to coordinate multiple processes or threads to access the critical resource (Yang, Jingjun, & Dongwen, 2009).

In the producer/consumer problem, some type of data (records or characters) are produced by one or more producers and these data are placed into a buffer. These items are taken out from the buffer by a single consumer. The system has to prevent the conflict of buffer operations. That is, accessing the buffer at any time has to occur by one agent (producer or consumer). It also makes sure that the producer can't add data into a full buffer and the consumer can't read data from an empty buffer (William, 2012) (Yang, Jingjun, & Dongwen, 2009).

The solution for the full buffer is to make the producer go to sleep until there is a room in the buffer. In the same way in an empty buffer, the consumer goes to sleep until the producer puts data into the buffer which makes the consumer wake up to consume the data (Yang, Jingjun, & Dongwen, 2009). Figure 4.7 shows the solution for the producer/consumer problem (Silberschatz, Galvin, & Gagne, 2018).

<pre> while (true) { . . . /* produce an item */ . . . wait(empty); wait(mutex); . . . /* add item to the buffer */ . . . signal(mutex); signal(full); } </pre>	<pre> while (true) { wait(full); wait(mutex); . . . /* remove an item from buffer */ . . . signal(mutex); signal(empty); . . . /* consume the item */ . . . } </pre>
A	b

Figure 4.7: Producer/Consumer Pseudo Code, (a) Producer Code (b) Consumer Code.

CHAPTER 5

OFFLINE METHODOLOGY AND RESULTS

5.1 Introduction

As discussed earlier a BCI system is constructed from stages such as data acquisition, preprocessing, feature extraction/selection and classification. The aim is to distinguish the motor discriminative features and extract it from the EEG signal. Researchers used a lot of methods for preprocessing stage. Some of them preferred to use a spectral method like Fourier transform or wavelet transform. Others used a temporal method such as filtering or even used a complex spatial method such as Independent Component Analysis (ICA) or Principal Component Analysis (PCA).

In real-time BCI system, reliable statistical machine learning schemes are required for modeling relationships between cognitive movements and high-dimensional neuronal features. Such kinds of methods have to be capable of operating efficiently with the fastest computational time (Tim, et al., November 2015).

Since, the aim of the thesis is trying to construct a real time BCI, so the tools used in this thesis have to be as simple as possible. The method simplicity has a fast response time which is a crucial demand in real time systems. This chapter focuses on offline BCI system to determine a proper preprocessing scheme and a robust statistical method to extract reliable EEG motor features. It also tries to determine the suitable classifier between the two most popular classifiers (MLP and SVM). First, a brief description will be provided for the EEG datasets used in this thesis. Then, the proposed method will be explained and the results will be given. Also, a second enhanced method with a detail exploration of its performance and feature selection procedure with its results will be introduced.

The programming language used to extract the results of this chapter is MATLAB R2015b operated on *Lenovo* laptop E550, core I7 with operation system Microsoft Windows 10.

5.2 EEG Datasets

This section provides a brief description of the dataset used in this thesis. It will give some details about important issues like number of the participated subjects, EEG tasks, number of EEG trials ..etc.

5.2.1 EMOTIV Dataset

Real right/left finger movements were performed in this dataset to acquire EEG signals. Figure 5.1 illustrates the movement procedure. Thirteen subjects participated to perform these two tasks. Before data acquisition, each subject filled and signed a consent form explaining the aim and the experimental procedure. EEG signals are acquired using a standard EMOTIV EPOC+ headset. *EmotivXavierControlPanel* software provided by EMOTIV Company was used for acquiring the EEG signal with sampling rate 128Hz. Microsoft Powerpoint also used to provide auditory stimuli. In each session, the subject relaxed in an armchair with closed eye listening to the auditory stimuli. These stimuli informed the subjects to start and stop their fingers. The duration of each movement was six seconds while the resting duration in between had different durations. Four movements were performed in each session. So, the size of the dataset is 13 subjects x 2 task x 4 trials x 14 channel x 768 sample. Figure 5.2 shows a sample of three seconds for EMOTIV dataset.



Figure 5.1: EMOTIV Dataset Real Finger Movements.

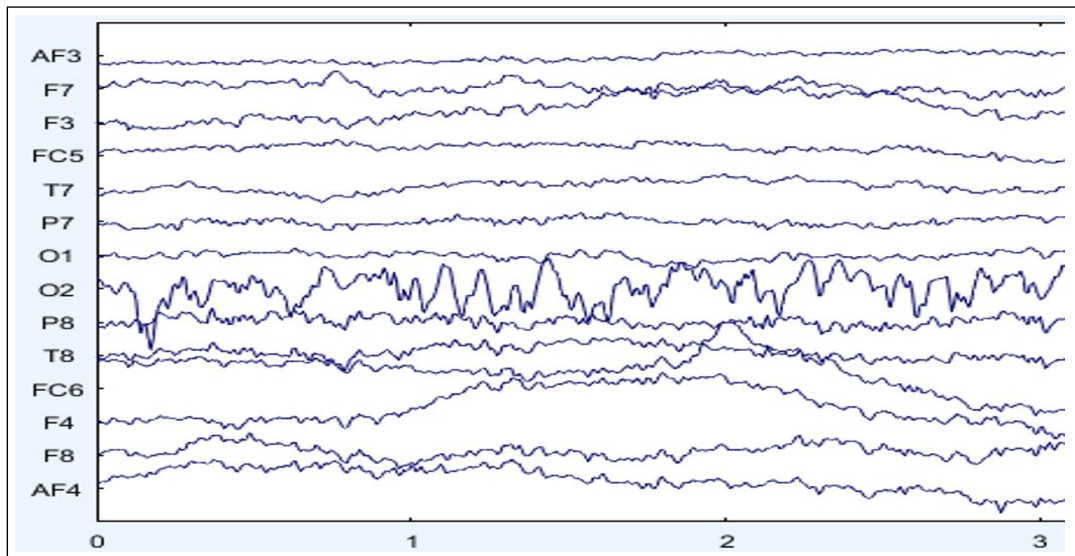


Figure 5.2: A Sample of EMOTIV Dataset

5.2.2 BCI Competition III Data set IVa

University Medicine Berlin provides EEG dataset called Data set IVa. This dataset is one of the different datasets uploaded to the internet under the name BCI Competition III. The aim of publishing this dataset was to evaluate the signal processing and classification methods used in BCI and compare its results with the previous competition datasets. In this dataset, five healthy subjects participate to acquire motor EEG imaginary signals. Three different imaginary tasks are recorded which are right/left hand and left foot imaginary movement. Only right hand and left foot movements were released in the website. The subject sat in a comfortable armchair staring on a screen in front of him and relaxing their arms on the arms of the chair. The subject starts to perform the predefined imaginary movement whenever a visual cue appeared on the screen. The cues presentation was randomly suspended by periods from 1.75 to 2.25 s to provide relaxation intervals for the subjects. 280 are the total number of trials recorded from the five subjects. Two sampling rates (1000Hz and 250Hz) is provided. In this thesis, the dataset of 1000Hz sampling rate was used. A standard electrode placement, called extended 10-20 electrode placement system of 118 electrodes was used to acquiring the EEG signal. Hence, the size of the dataset is 280 trials x 2 task x 118 channels x 3000 sample. Figure 5.3 shows three seconds sample of 118 channel for this dataset (Müller & Blankertz, 2004).

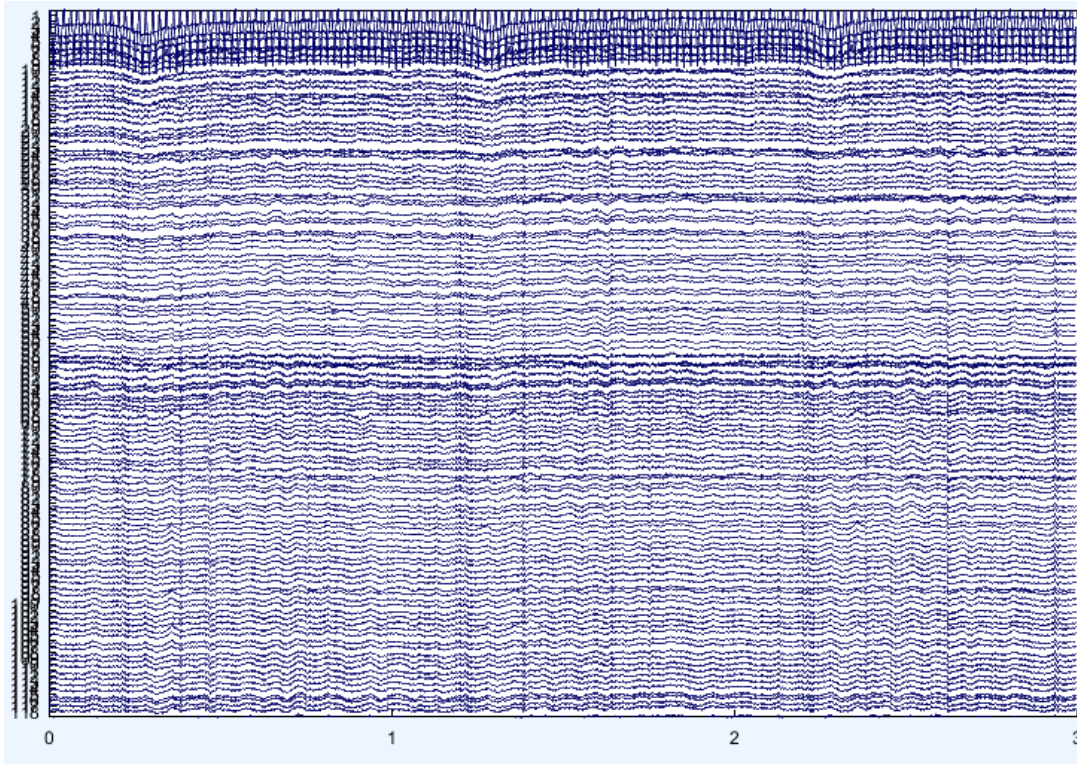


Figure 5.3: A Sample of BCI Competition III Dataset IVa

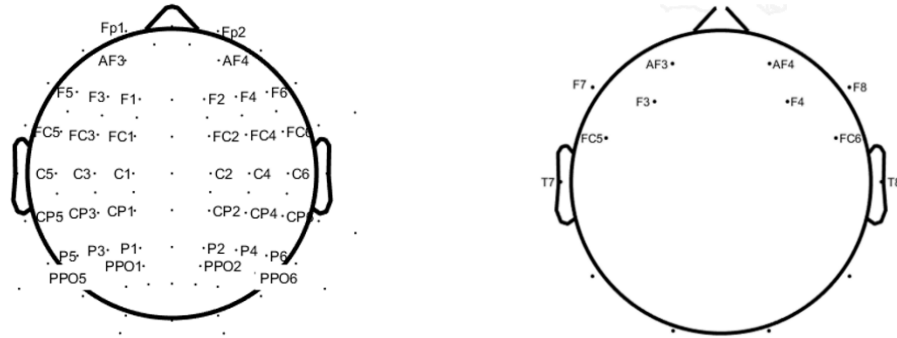
5.3 Classifiers Settings

This thesis used MLP classifier of one hidden layer using *Levenberg-Marquardt backpropagation* training algorithm with error rate set to 0.001 and mean square error as an evaluation function. The output layer is constructed from a log-sigmoid transfer function and two output nodes. SVM used the *radial base kernel function* with the *KernelScale* set to auto. Since, the available EEG datasets have a limited size so, 10-fold cross-validation was used in this chapter to evaluate the performance of both classifiers.

5.4 Proposed Method

The method consists of six stages. These stages try to extract the most discriminative features from the EEG motor signal and classify it. The suggested method is described as follows:

- **EEG subtraction:** Subtract the subject EEG motionless signal from its EEG motor signal.
- **Channel selection:** Selecting the EEG channels that are most related to the motor signals. Figure 5.4 shows the selected EEG motor channels.



(a) Selected channels of BCI datasets (b) Selected channels of EMOTIV datasets

Figure 5.4: The Selected EEG Motor Channels for The Both Datasets.

- **Band extraction using DWT:** Decomposing the EEG signal using DWT to extract the level 3 details coefficients. These coefficients represent the high Beta and Gamma rhythms for the analyzed datasets.
- **Cross correlation of the effective channel with right/left hemisphere:** Two channels (F4 and F3) were selected to be the effective channels. The selection of these two channels is based on its anatomical locations. These locations lie upon the brain regions or near the brain regions that generate the EEG motor signals. In this stage, reference signals (RF and LF) were constructed from the two effective channels (F4 and F3) respectively. These reference signals are the average signals of the effective channels for the training trials only. Then, these two reference signals (RF and LF) are cross-correlated with the right/left hemisphere channels respectively and this was performed for the all trials in the dataset.
- **Statistical parameter calculation:** Extraction of normalized features from the previous stage signals using by ten statistical tools (1st quartile, 3rd quartile, entropy, mode, mean, median, SD, max, min, and range).
- **Classification:** The extracted features were classified using two classifiers (MLP and SVM).

Figure 5.5: shows the block diagram of the proposed method (Al- Dabag & Ozkurt, 24 October 2018).

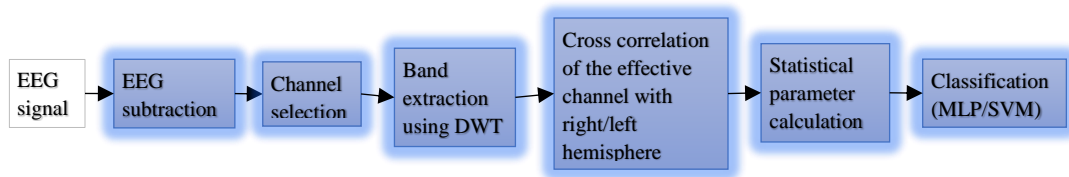


Figure 5.5: The Proposed Method Structure.

5.4.1 Experiments and Results

The proposed method was applied using both datasets mentioned previously. The Db3 wavelet filter is applied to signal and the detail coefficients from level 3 of DWT is used. This constructs Gamma and high Beta bands from BCI competition and EMOTIV datasets respectively. The experiments of the proposed method were grouped into two procedures which are called patient based classification and movement based classification. In the patient based classification, the classification of each subject was done individually. In the second procedure, the mixed features of all subjects are classified together to know the reaction of the proposed method whenever classifying the mixed features from different subjects.

Each EEG trial in EMOTIV dataset was segmented into 6 segments of one-second so the total trials of each subject is 24 trial. One of a resting period was saved for EEG subtraction. Hence, the total size of EMOTIV dataset became 13 subject x 2 task x 24 trial x 14 channel x 128 sample with one resting signal of 1 second. Each trial of BCI competition dataset was also segmented into movement and resting signals but its period was left unchanged just as provided in the website (Al- Dabag & Ozkurt, 24 October 2018).

5.4.2 Patient-Based Classification

In MLP classifier, the number of the neurons in the hidden layer has to be determined experimentally. Hence, several experiments accomplished to determine the number of neurons that provide the best classification rate (best accuracy with low SD.).

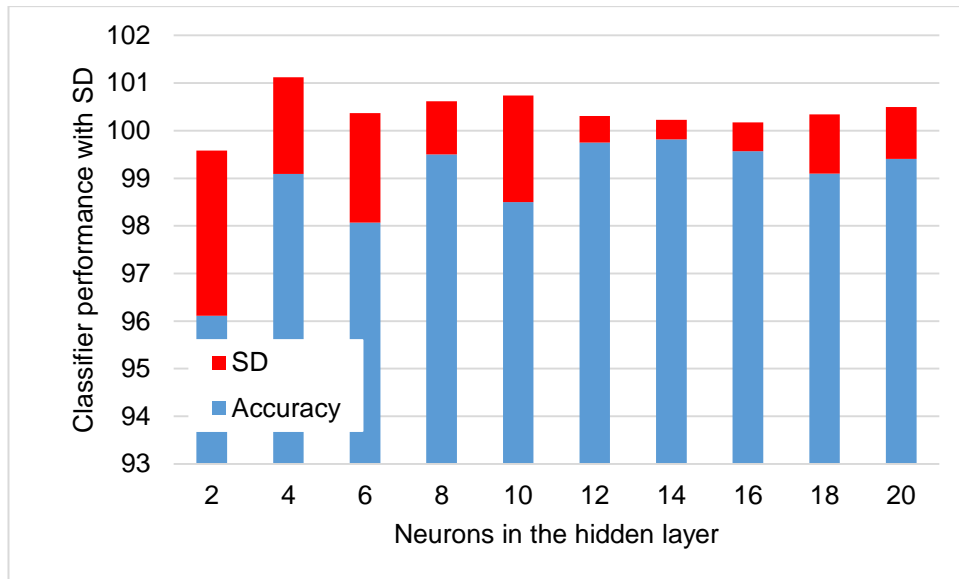


Figure 5.6: Patient Based Classification Experiments to Determine The Number of Nodes in The Hidden Layer for BCI Competition Dataset.

In BCI competition III IVa, this was done by varying the hidden layer neurons from 2-20 with step size of 2. Then, calculating the average of the five subject classification accuracies and its SD. Figure 5.6 shows the MLP performance with the different number of neurons in the hidden layer. This figure shows the average classification rates of five subjects (blue color) with SD. (red color). Clearly, MLP with 14 nodes in the hidden layer has the best performance.

For a better estimation of the proposed method, each classification was repeated 10 times using 10-fold cross-validation method to assess both classifiers (MLP and SVM). MLP was reconstructed using 14 nodes in the hidden layer. According to the described estimation procedure, MLP had variable performance due to the nature of the algorithm while SVM had a stable performance trend. Figure 5.7 shows a performance comparison between MLP and SVM classifications. This figure shows the best, the worst and the average of ten runs of MLP classifier for each subject. The results reflect that SVM had a better result for all subjects except for subject *aa* and *al* (Al- Dabag & Ozkurt, 24 October 2018).

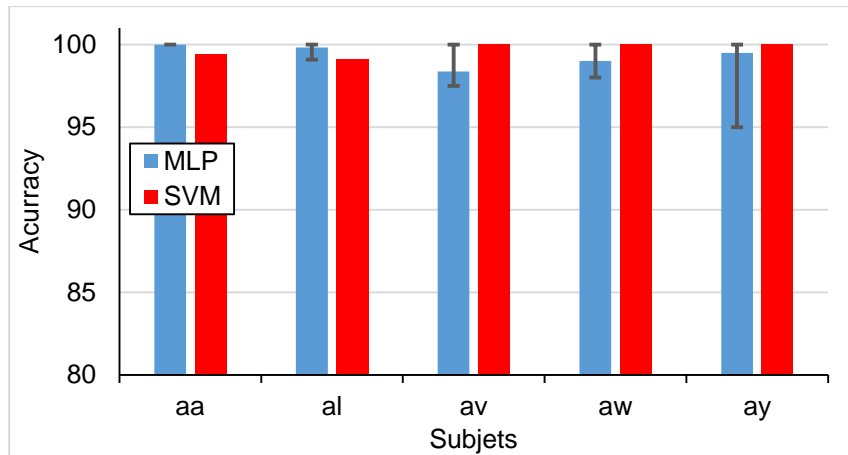


Figure 5.7: Patient Based Classification of BCI Competition III VIa Dataset.

In EMOTIV dataset, the same procedure was applied to determine the number of the neurons in the hidden layer of MLP. Figure 5.8 shows the performance of the MLP with different numbers of nodes in the hidden layer. The experiment changed the number of the nodes in the hidden layer from 2 to 24 with step size of 2 to determine the number of nodes that provides the best performance. This figure shows the SD, as well as the average of 13 subject classifications of each experiment. From the figure, it is obvious that twelve nodes in the hidden layer provided the best performance (Al-Dabag & Ozkurt, 24 October 2018).

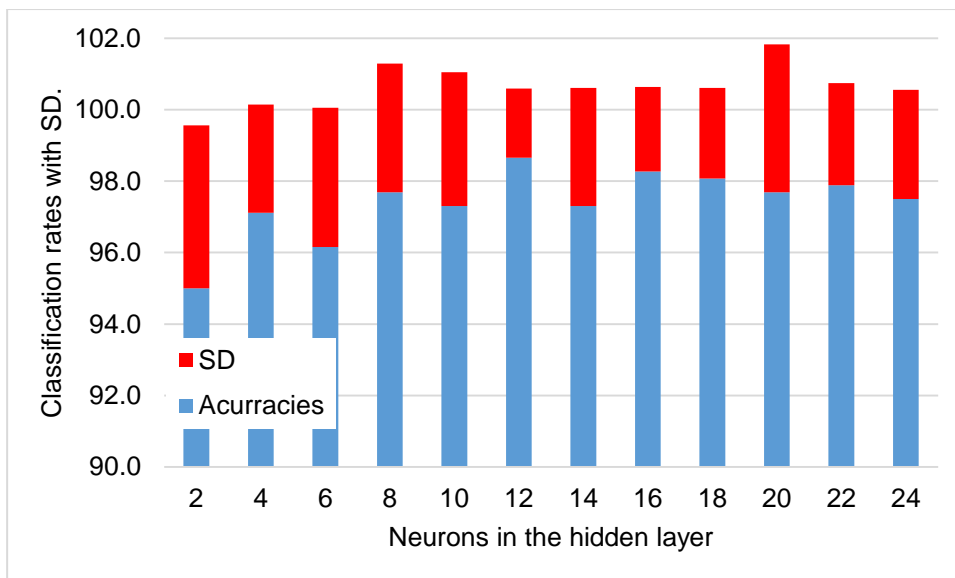


Figure 5.8 : Patient Based Classification Experiments to Determine The Number of Nodes in The Hidden Layer for EMOTIV Dataset.

The same estimation procedure was also applied to estimate the behavior of the two classifiers. Figure 5.9 shows both classifier performance for each subject in EMOTIV dataset. This figure clearly shows that the SVM classifier had for a majority the best classification rates (Al- Dabag & Ozkurt, 24 October 2018).

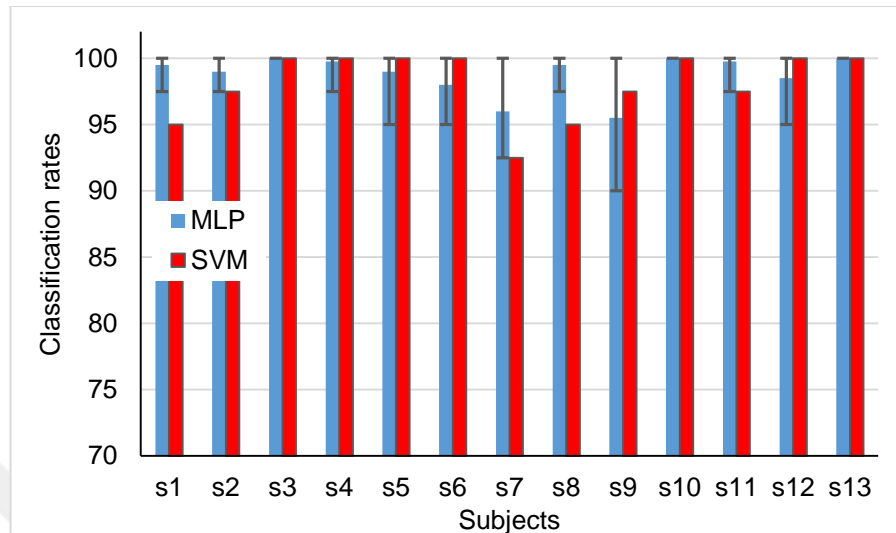


Figure 5.9 : Patient Based Classification Of EMOTIV Dataset.

For better comparison of both classifiers performance in a different dataset, the average of the all subject classification rates was calculated to evaluate each classifier performance. Figure 5.10 shows that both classifiers have classification rates above 98% for the different dataset.

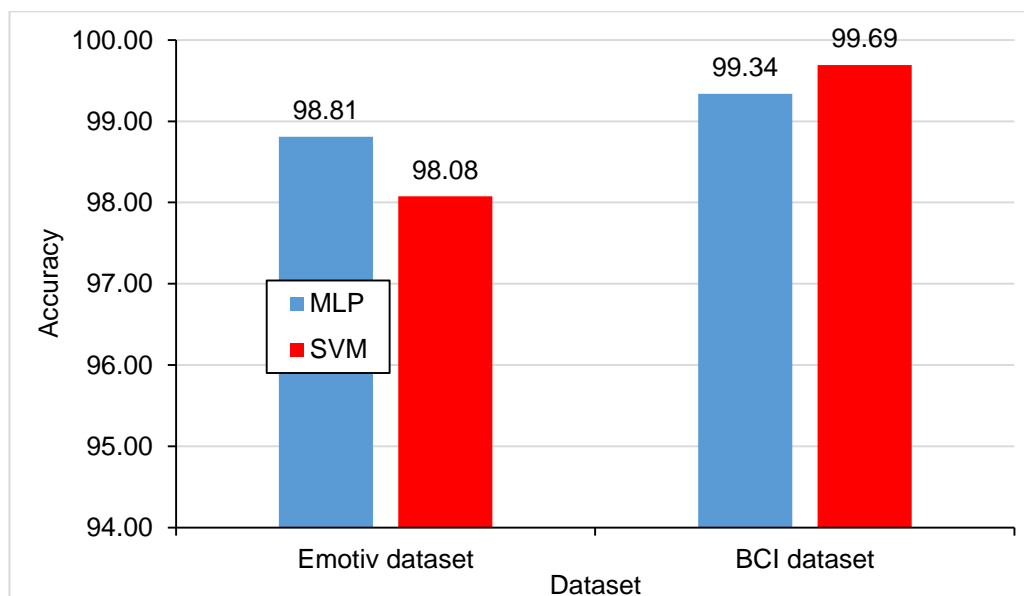


Figure 5.10: The Average of All Subject Accuracies in Both Datasets for Both Classifiers.

5.4.3 Movement Based Classification

As mentioned earlier, this procedure processed each subject individually using the proposed method and mixed the all subject features together and classify it. In the BCI competition III dataset, 186 training and 92 testing trials available after mixing all the subject features. A similar experimental procedure was done to determine the number of nodes in the hidden layer. Figure 5.11 shows MLP experiments with different number of nodes in the hidden layer. It is obvious that from 10 to 20 nodes in the hidden layer have a competitive performance. 10 nodes was chosen because it has less MLP structure complicity and smaller standard deviation than the rest (Al- Dabag & Ozkurt, 24 October 2018).

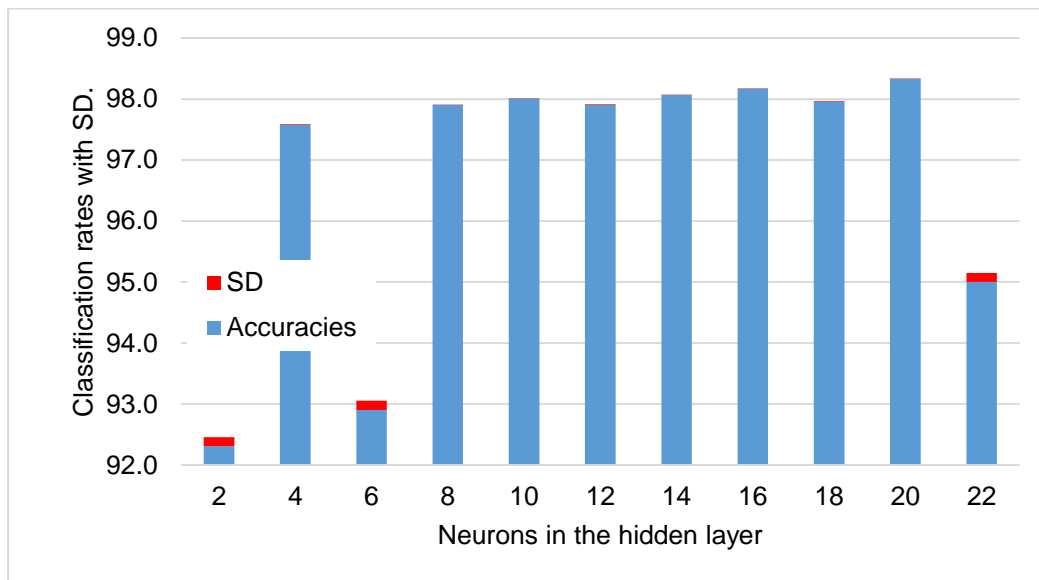


Figure 5.11 : Movement Based Classification Experiments to Determine The Number of Nodes in The Hidden Layer for BCI Competition Dataset.

In the EMOTIV dataset, there were 176 training and 88 testing trials after mixing the features of 13 subjects. Figure 5.12 shows the experiments to determine the efficient MLP structure. Twenty six nodes in the hidden layer were chosen since it has the best performance (Al- Dabag & Ozkurt, 24 October 2018).

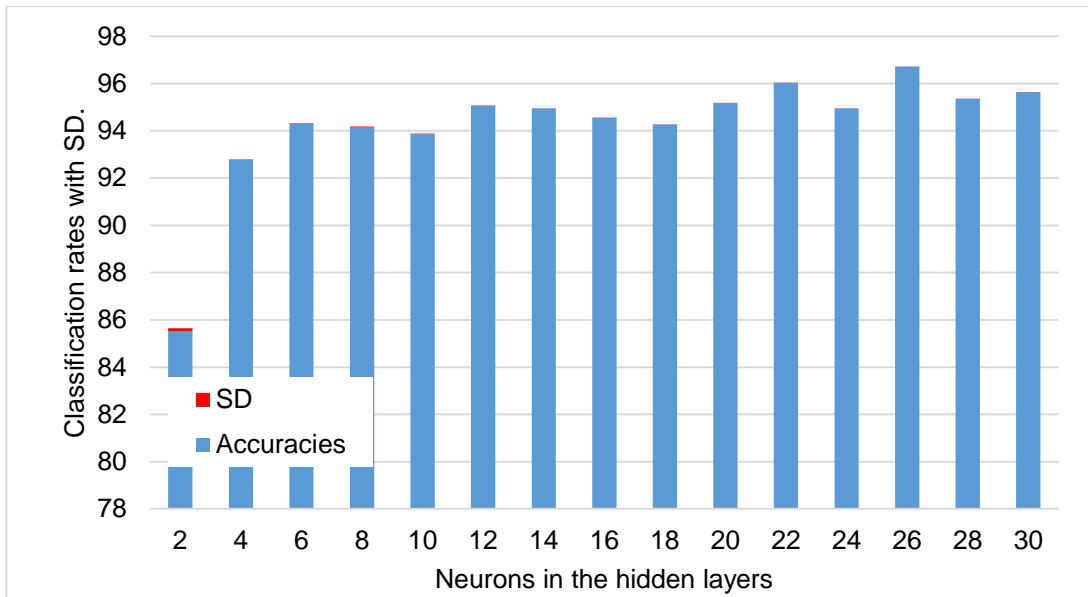


Figure 5.12: Movement Based Classification Experiments to Determine The Number of The Nodes in The Hidden Layer for EMOTIV Dataset.

The holdout estimation method with ten runs were performed to estimate the movement based classification for both datasets. Figure 5.13 shows the two classifiers behavior for both datasets. This figure shows that the classifiers still give an accuracy of above 92% even after merging features of different subjects (Al- Dabag & Ozkurt, 24 October 2018).

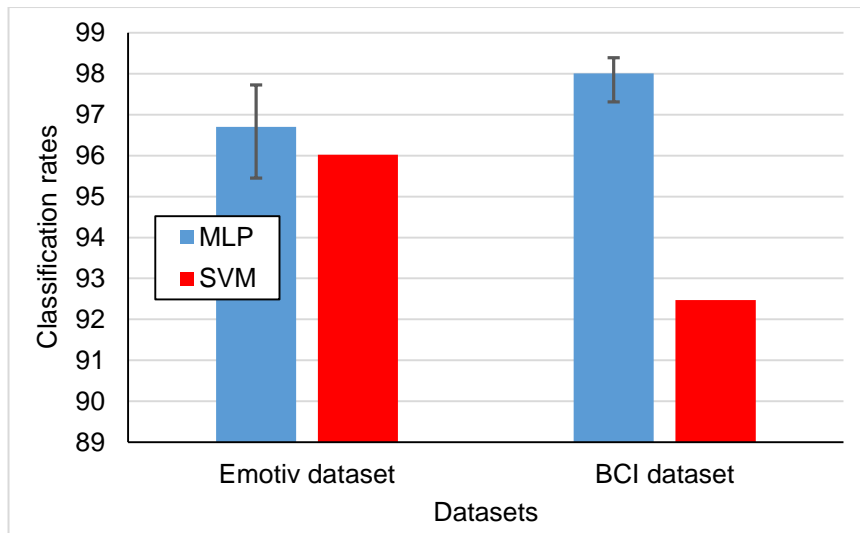


Figure 5.13: The Performance of Both Classifier Using Movement Based Classification.

5.4.4 Comparison with Previous Studies

The robustness of the proposed method was illustrated in Table 5.1. This table shows the classification rates using several methods compared with the proposed method and these methods were used the same dataset which is BCI competition III dataset IVa. The first method that has a competitive performance with the proposed method uses PCA with cross-covariance for preprocessing. The complexity of using PCA makes it relatively inappropriate for a real time system. The average performance of all methods had smaller performance compared with our proposed method. Furthermore, the proposed scheme does not employ complex methods.

Table 5.1 : Performance of Several Methods for Classifying Imaginary Motor EEG of BCI Competition III Dataset IVa.

Methods	Training/testing partition	Classification accuracy rate (%)					
		aa	al	av	aw	ay	mean
The proposed method MLP	10 fold cross validation 10 times	100±0	99.8±0.31	98.3±0.8	99±1.0	99.5±1.5	99.33
The proposed method SVM	10 fold cross validation	99.37	99	100	100	100	99.69
PCA-CCOV based MLP, SVM (Zarei, He, Siuly, & Zhang, 2017)	50/50	99.15	100	100	100	99.15	99.66
LA-DE-SVM (Bhattacharyya, Sengupta, Chakraborti, Konar, & Tibarewala, 2013)	10 fold cross validation	97.06	100	100	100	100	99.41
OA-NB (Siuly & Zhang, 2016)	10 fold cross validation	97.92	97.88	98.26	94.47	93.26	96.36
CS-SVM (F.Ince, Goksu, H.Tewfik, & Arica, 2009)	10 fold cross validation	95.6	99.7	90.5	98.4	95.7	96
CC-LS-SVM (Siuly & Li, 2012)	10 fold cross validation	97.88	99.17	98.73	93.43	89.36	95.72
STFSCSP-WNBC (Miao, Zeng, Wang, Zhao, & Liu, 2017)	10 fold cross validation 10 times	81.25	100	65.31	93.30	92.06	86.38

5.5 The Enhanced Method

Further experiments for the proposed method shows the need for a noise reduction step. Since the resting period may be corrupted by noise so subtracting it from the motor EEG signal may add noise to the signal rather than throwing out the baseline signal. Hence, another denoising stage is proposed.

EEG signals are interfered with a lot of artifacts as mentioned in Chapter 2. The most harmful artifacts are the ocular and muscular movement artifacts. These artifacts have a large amplitude and low frequencies in contrast with the motor EEG signals. Hence, using traditional thresholding techniques (soft and hard thresholding) cannot be used in this application. The goal is to remove the large amplitude and low frequencies which is the opposite operation of hard and soft thresholding. The following equation illustrates our thresholding technique which is applied to the second level of the DWT details coefficients.

$$x_d(n) = \begin{cases} x(n) & |x(n)| < T \\ T & |x(n)| \geq T \end{cases} \quad (1)$$

where T is the threshold which is equal to $\log(\text{length}(x))$

DWT Symlet filter (sym8) used for decomposing the EEG motor signal. The denoising technique replaced the first three stages in our proposed method.

There is also a simple modification in the fourth stage (*Cross correlation of the effective channel with right/left hemisphere*). In this stage, there is no need to construct a reference signal (RF and LF). Hence, the instantaneous effective channel (F4 and F3) of each trial are cross-correlated with the right/left hemisphere channels respectively. The following stages illustrate the enhanced method.

- **Normalization:** this stage scales the EEG signal to make it lie between 1 and -1 by dividing each EEG channel by its maximum absolute value.
- **EEG artifact removal:** EEG denoising using the thresholding technique stated in equation 1 to denoise the second level of the DWT details coefficients.
- **Cross correlation of the effective channel with right/left hemisphere:** emphasizing the EEG motor signal by cross-correlation of the effective channel (F4 and F3) of each trial with the right/left hemisphere channels.

- **Statistical parameter calculation:** the same ten statistical features (min, max, mean, mode, median, SD, range, entropy, 1st quartile and 3rd quartile) are used to extract effective features from EEG signal.
- **Classification:** MLP and SVM are used for classification the constructed features.

Figure 5.14 shows block diagram of this the method. (Al-Dabag & Ozkurt, June 2018).

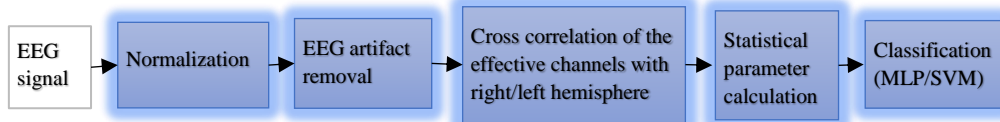
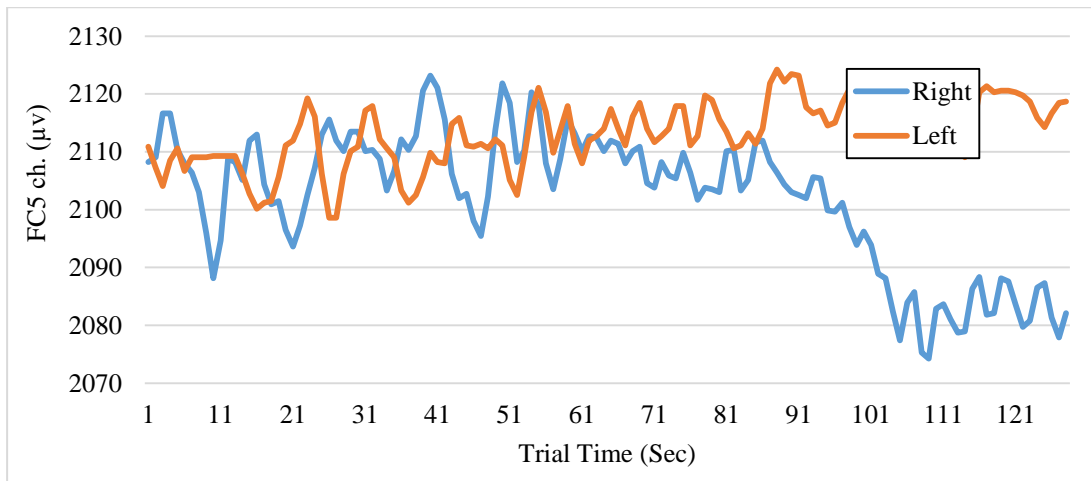
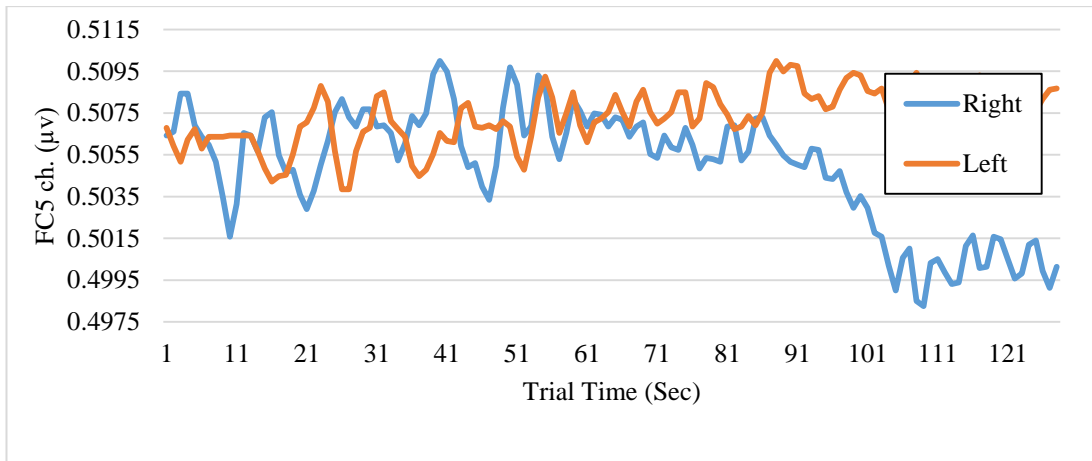


Figure 5.14: The Enhanced Method Block Diagram.

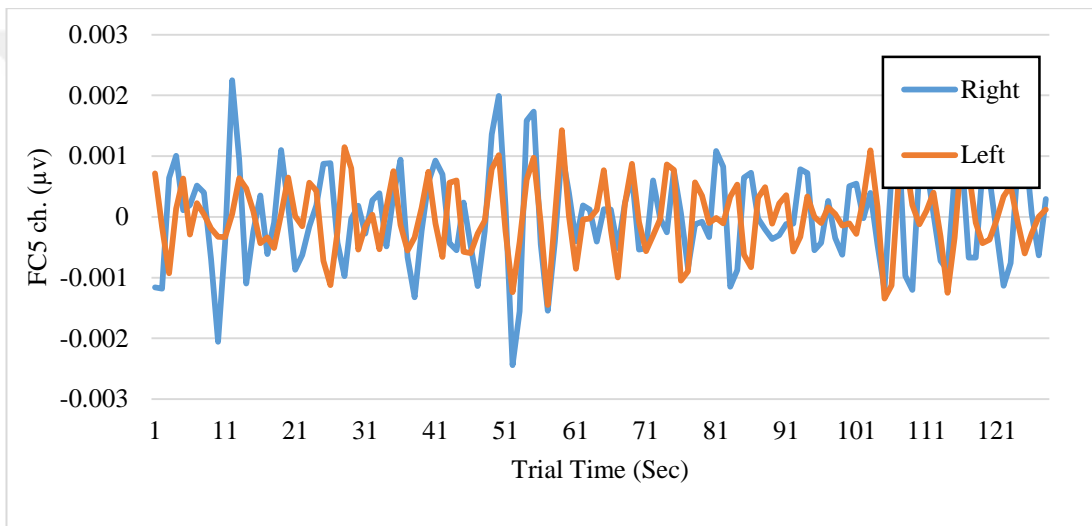
Figure 5.15 shows the effect of this method during its' operation on a sample of FC5 channel and this is shown for both tasks. Figure 5.15 (a) shows the original EEG signal and it is clear from the EEG signal, there is an artifacts that corrupt the both tasks. Figure 5.15 (b) shows the same signal after normalization. Figure 5.15 (c) shows the EEG signal moving around the baseline and the artifacts was removed. The effect of cross-correlation was shown in Figure 5.15 (d) and it obviously clear that the difference between the both tasks was enlarged but the both signals are also enlarged. The statistical stage overcome the cross-correlation stage by feature extraction and clearly there are more than one statistical method that discriminate the both tasks such as range, mode... etc., see Figure 5.15 (d)



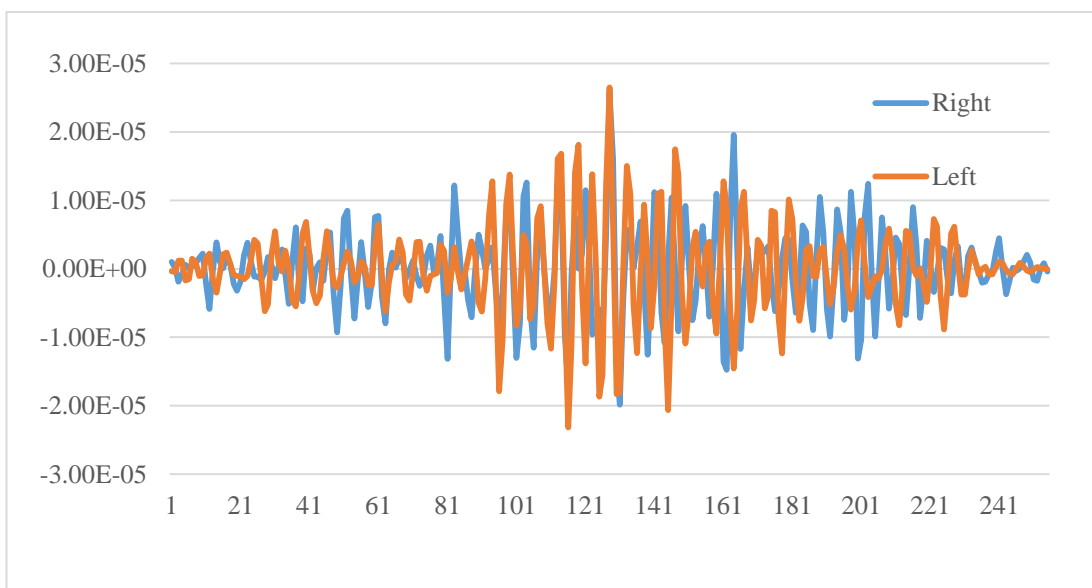
(a) Original signal



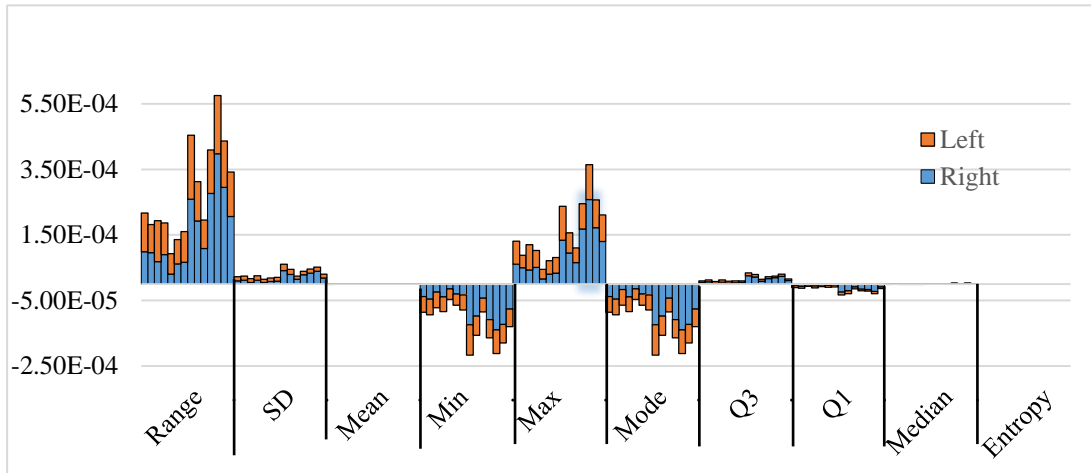
(b) Normalized signal



(c) After artifact removal



(d) After cross-correlation



(e) After feature extraction

Figure 5.15 : The Stages Performance of The Enhanced Method on FC5 Channel for Both Tasks (Right/Left Finger Movement)

5.5.1 Experiments and Results

The aim of this thesis is to build real time BCI based on EMOTIV EPOC+ headset so the enhanced method was applied only on EMOTIV dataset. Since, other datasets were recorded using different amplifiers, electrodes and even in different environments so processing only the EMOTIV dataset will reduce the number of variables processed by the enhanced method. Each epoch in the dataset overlapped 32 samples with each other to increase the number of trials in the dataset so the size of the dataset became 13 subject x 2 task x 86 trial x 14 channel x 128 sample. The same setting of the classifier was used with the same estimation method (10-fold cross-validation). The following sections first will explore some stages of the enhanced method to view its operation. Then, it will explore the classification rates before and after features selection.

5.5.1.1 EEG artifacts removal

The effect of applying the EEG artifact removal stage is shown in Figure 5.13. Figure 5.16 (a and c) show the power spectrum of the Alpha band before denoising and it is clear that there is no spectral difference between the two hemispheres. Figure 5.16 (b and d) show the power spectrum of the same band after denoising. In the last two figures, the difference between the two hemispheres after denoising is emphasized and this phenomenon has an opposite reaction between the two tasks (Al-Dabag & Ozkurt, June 2018).

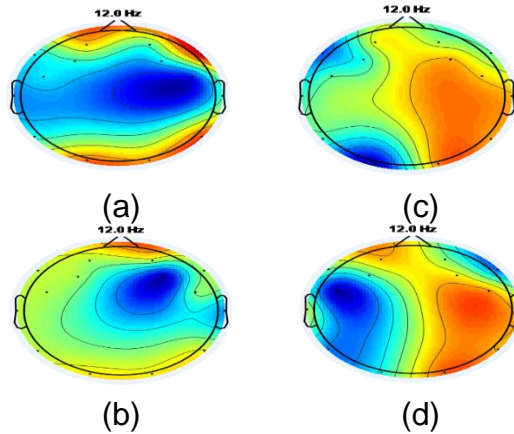


Figure 5.16: EEG Topography of Alpha Band Before And After Applying Eeg Artifacts Removal. (a) Original EEG Right Movement. (b) Denoised EEG Right Movement. (c) Original EEG Left movement. (d) Denoised EEG Left Movement.

5.5.1.2 Cross correlation stage

The two effective channels (channels F4 & F3) cross-correlation with the right/left hemisphere channels enlarge the temporal magnitude difference between the two hemispheres, see Figure 5.17 In this figure, the left and right hemisphere channels are

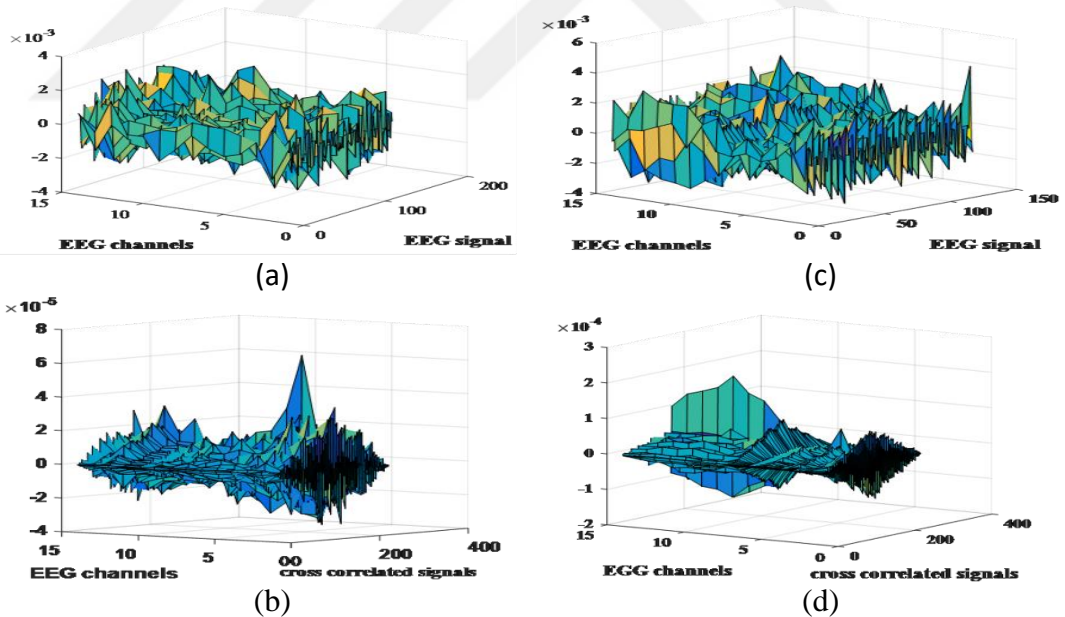


Figure 5.17: Effects of Cross Correlation Stage with Effective Channels. (a) Right Movements Channels Before Cross Correlation. (b) Right Movements Channels After Cross Correlation. (c) Left Movements Channels Before Cross Correlation. (d) Left Movements Channels After Cross Correlation.

numbered as 1-7 and 8-14 respectively. Figure 5.17(a) and (b) shows the EEG signal before and after the cross correlation. Comparing these two figures shows that this

stage enlarges the signal magnitude in the left hemisphere (channels 1-7). In contrast, opposite behavior is observed in left finger movement (Figure 5.17 c and a).

5.5.1.3 Classification

The number of the hidden layer nodes of MLP have to be determined before the evaluation of the classification rates. Therefore, several experiments were also performed to determine the number of the node by varying it from 2 to 28. Figure 5.18 shows these experiments. This figure shows that twenty nodes in the hidden layer have the best classification rate with less standard deviation. Hence, this number of neurons was adopted to reconstruct the MLP (Al-Dabag & Ozkurt, June 2018).

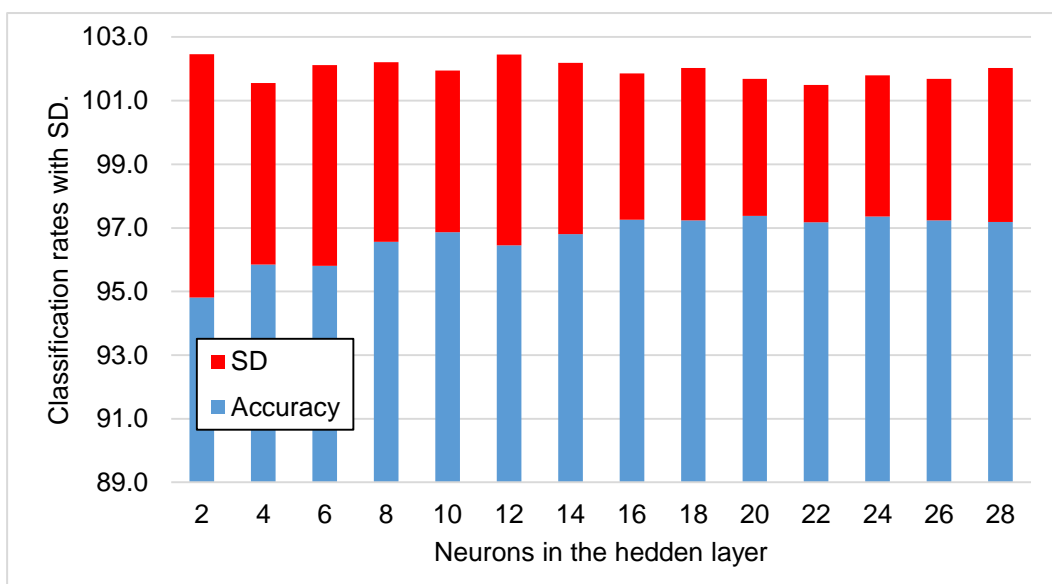


Figure 5.18: MLP Performance Evaluation with Different Number of Nodes in The Hidden Layer.

Ten runs were also used as well as 10-fold cross-validation to estimate the performance of the two classifiers. Figure 5.19 shows the classification rates of both classifiers for all subjects. This figure shows that there is a competitive behavior between the two classifiers. To clarify the two classifiers performance, Figure 5.20 shows the average classification rates of two classifiers for all subjects. It is clear these two classifiers have similar behavior (Al-Dabag & Ozkurt, June 2018).

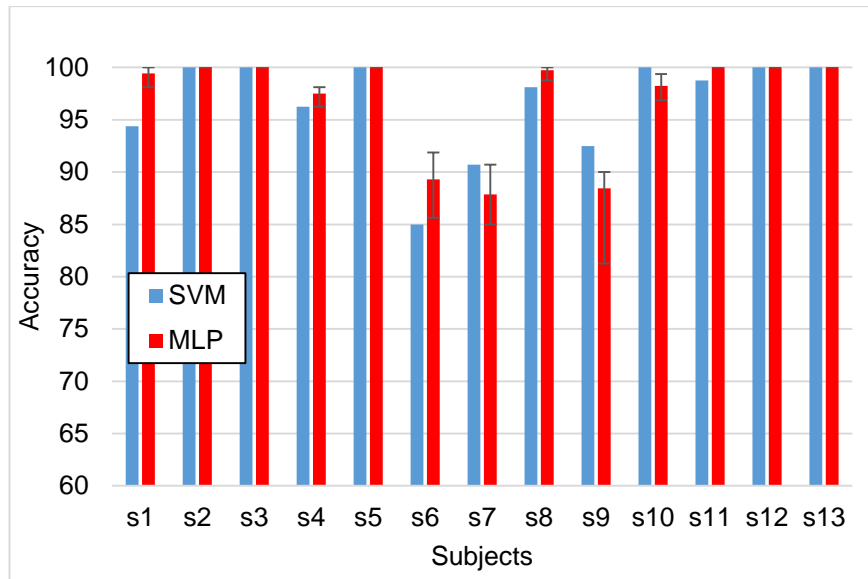


Figure 5.19: Classification Rates of Both Classifiers for All Subjects.

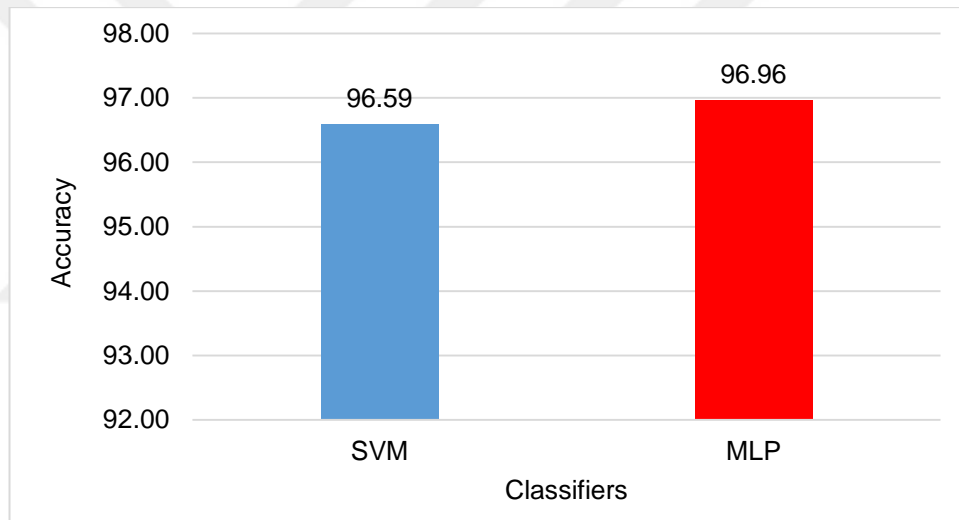


Figure 5.20: The Average Classification Rates of Both Classifiers for All Subjects.

5.6 Feature selection and classification

Genetic algorithm was used to search for the best statistical parameter calculation methods. The population size was chosen to be 20 since diversity is ensured and reducing the harmful effects of the mutation operator. The enhanced method with SVM were used in the GA fitness function to evaluate each GA individual. Since the GA cost value consists of two quantities; classification rates and the number of unused statistical methods. Therefore, the cost (fitness) function had 99% for the average of thirteen subject classification rates and 1% for the unused statistical methods. GA needed eight iterations to find the best statistical parameters because the classification

rates had good performance in advance. It is found that only three statistical methods (mode, max, and SD.) were enough to discriminate EEG signal properly among ten methods.

After the best statistical methods were found by the GA., the amount of EEG features are reduced from 140 features (14 EEG channel x no. of statistical methods) to only 42 features. Table 5.2 illustrates the percentage ratios of EEG features before and after feature selection (Al -Dabag, Ozkurt, & Al-Aimam, 4-6 October 2018).

Table 5.2: EEG Features Ratios Before And After Features Selection.

	Data Amount	140 features	42 features
Original data	14 channels x 128 samples	7.8%	2.3%
Cross correlated Data	14 channels x 255 samples	3.9%	1.1%

Let's examine the classifier's performances before and after features selection. First, the best statistical methods are used in the enhanced method with SVM to process and classify the EMOTIV dataset. Figure 5.21 shows the classification rates before and after feature selection using SVM classifier. The figure shows that the overall classification rates after features selection are enhanced or have similar classification rates before features selection except in subject seven.

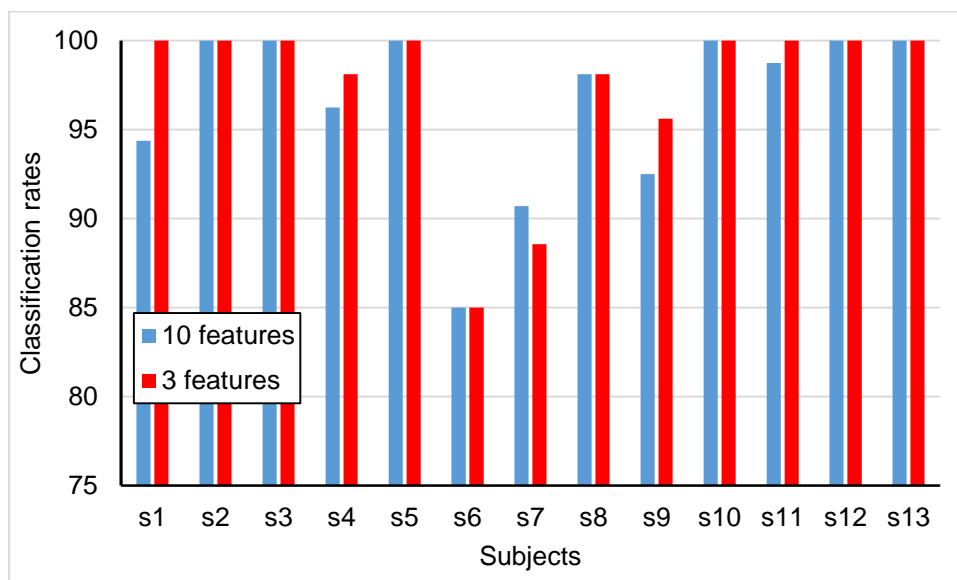


Figure 5.21: SVM Classification Rates Before And After Feature Selection for 13 Subjects.

Secondly, a similar comparison is made for MLP classification rates before and after features selection and this comparison uses the same number of neurons in the hidden

layer (20 nodes). Figure 5.22 shows this comparison. In this figure, the overall classification rates had similar or enhanced values especially in subjects (7 and 9) except in subject 6 had limited degradations (Al -Dabag, Ozkurt, & Al-Aimam, 4-6 October 2018).

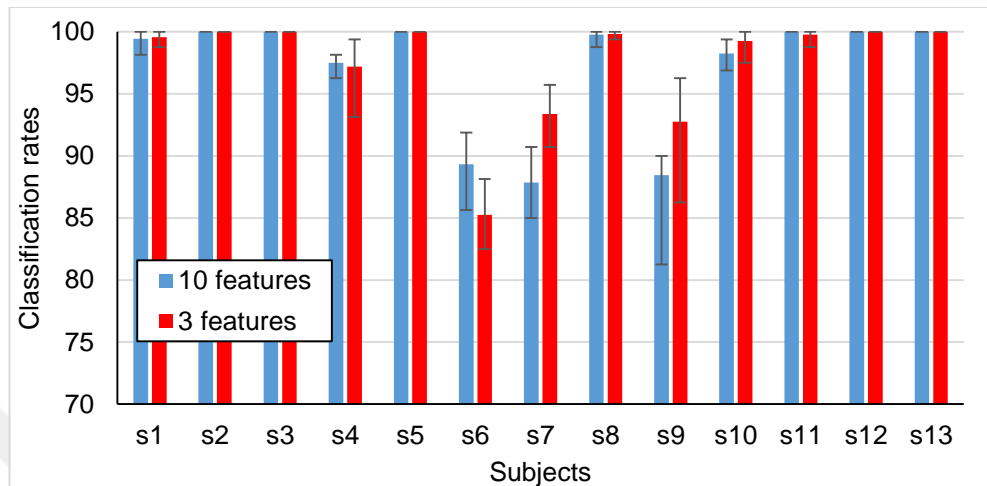


Figure 5.22: MLP Classification Rates Before And After Features Selection.

Since the number of statistical methods are reduced so let's examine its effects on the architecture of MLP (number of nodes in the hidden layer). Therefore, other experiments were performed by changing it from 2 to 20 nodes. Figure 5.23 illustrates the results produced by these experiments. It is clear that 14 nodes in the hidden layer has better performance than others. Figure 5.24 shows the classification rates for all subjects using two different structures of MLP. This figure shows a similar behavior of the two structures of MLP but one of them has less structure complicity (14 nodes in the hidden layer of MLP) (Al -Dabag, Ozkurt, & Al-Aimam, 4-6 October 2018).

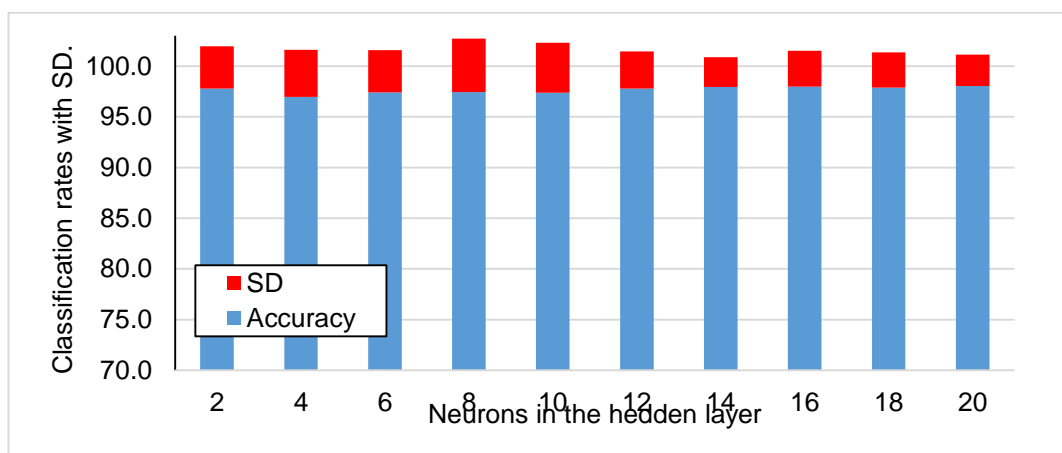


Figure 5.23: MLP Classification Rates with SD Using Different Number of Neurons in The Hidden Layer.

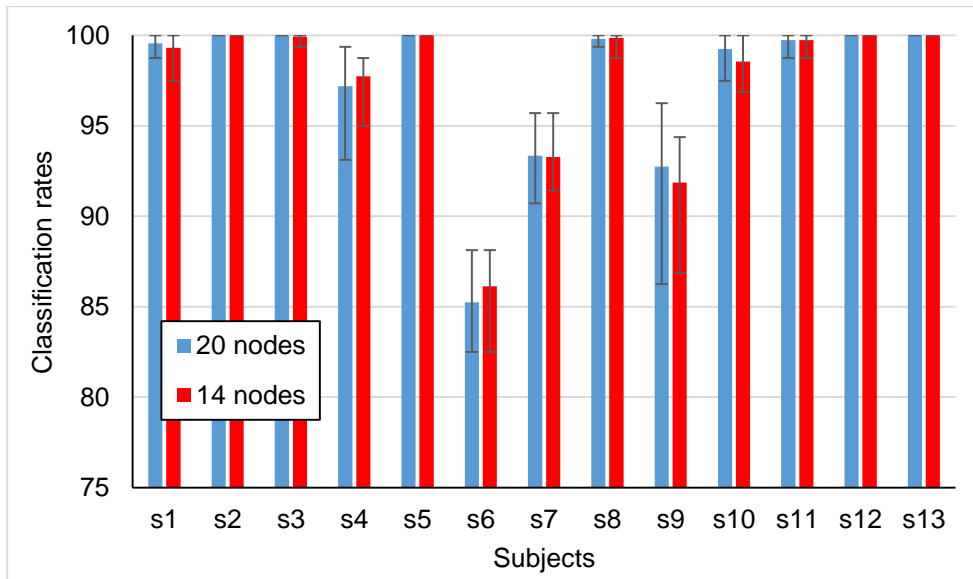


Figure 5.24: MLP Performance Using Two Different Architectures.

The average of thirteen subject classification rates for the redundant features and the discriminative features were calculated to summarize the performance of the two classifiers. Figure 5.25 illustrates the SVM performance enhancement before and after features selection. Figure 5.26 also illustrates this enhancement in MLP classifier before and after feature selection (Al -Dabag, Ozkurt, & Al-Aimam, 4-6 October 2018).

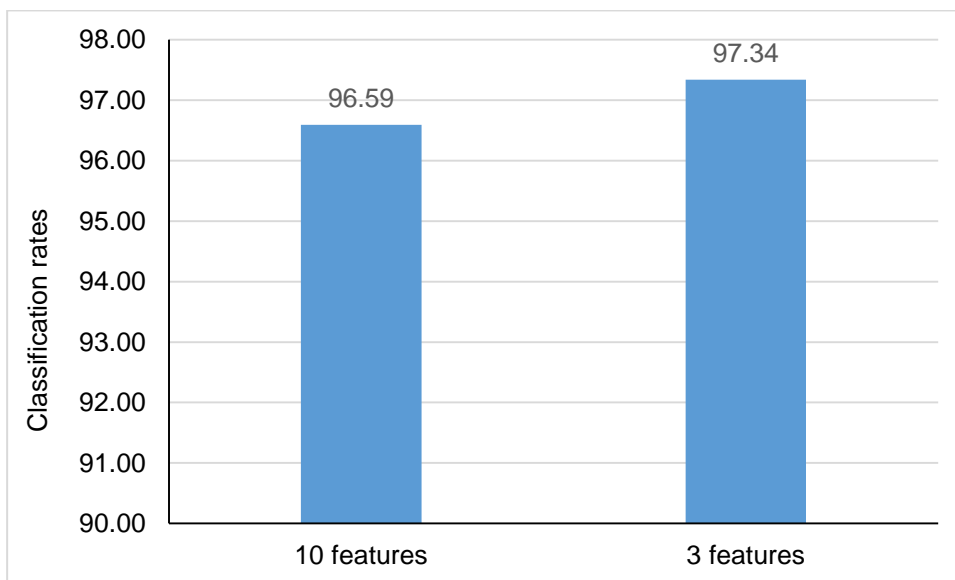


Figure 5.25: The Average of SVM Classification Rates Before And After Features Selection.

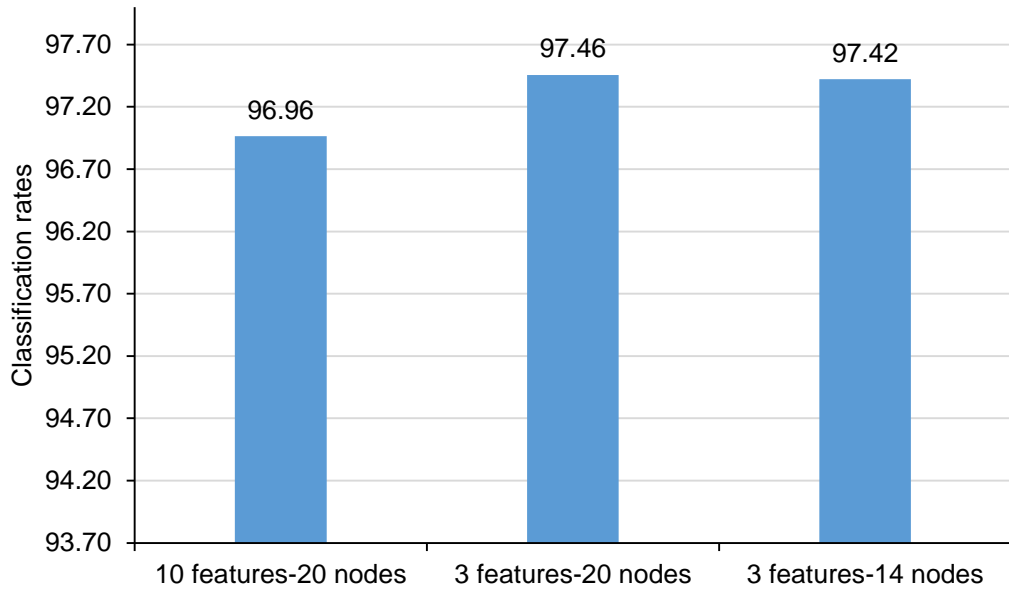


Figure 5.26: The Average of MLP Classification Rates Before And After Feature Selection of Two Different MLP Structure.

The classifier processing time is another parameter that has to be examined to help us for selecting the best classifier so the computation time of the enhanced method with the classifier has been measured before and after feature selection. Figure 5.27 illustrates the processing time of SVM using the best features extraction methods and the ten statistical methods. This figure shows that getting rid of the redundant features has another effect which is reducing the processing time of the classifier.

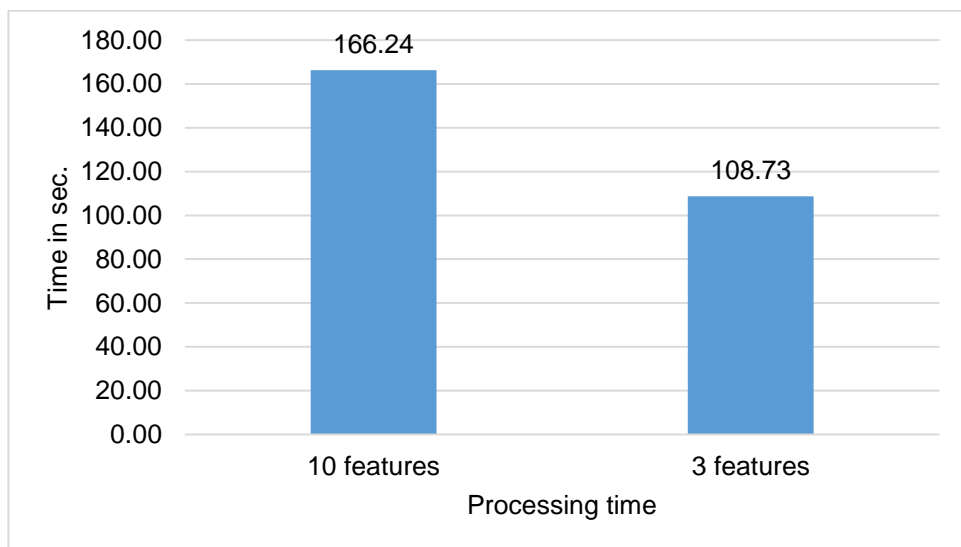


Figure 5.27: Processing Time of SVM Using Two Different Number of Statistical Methods.

Let's examine the behavior of different architecture of MLP described earlier. Figure 5.28 shows the processing times of MLP using different architectures and different amount of features. The figure illustrates that the processing time is also reduced as a consequence of classifying only the discriminative features and this performance had further enhanced by using the simplest MLP architecture (Al -Dabag, Ozkurt, & Al-Aimam, 4-6 October 2018).

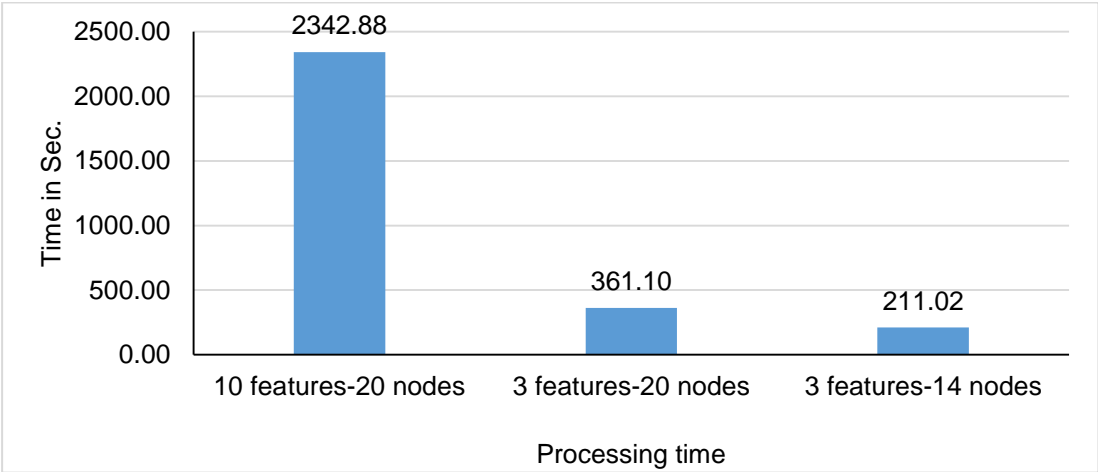


Figure 5.28: MLP Processing Time of Different Architectures.

CHAPTER 6

ONLINE METHODOLOGY AND RESULTS

6.1 Introduction

The earlier chapters focus on the EEG analysis and artifact removal methods to classify the performed cognitive EEG signals. Those studies are used an offline dataset of 13 subjects to find a simple and appropriate methodology to predict their movements. Building an interactive BCI is a challenging task since it should predict the performed task fast and accurately. Thus all stages of BCI system should be implemented effectively.

This chapter provides an online BCI prototype that acquires, preprocesses and classifies EEG raw data. This online BCI is based on the enhanced method mentioned in the previous chapter. The proposed online BCI is a software written using two programming languages (C# and MATLAB). The hybrid BCI software makes use of the facilities provided by the two programming languages and this leads to improve its performance and flexibility. This chapter focuses on the challenges faced during EEG acquisition stage rather than other stages of BCI system. It introduces an efficient EEG acquisition procedure which gets EEG raw signals and delivers it to the next BCI stage at the real time.

The next section demonstrates the experimental procedure. Then, how Matlab preprocessing, feature extraction and classification software is combined with EMOTIV data acquisition software. Section 6.4 provides the way of EMOTIV configuration for direct acquiring the EEG raw signal. Section 6.5 explores BCI software and its graphical user interface (GUI). Finally, the last three sections explore the response time, the classification rates obtained in the experiments and a brief discussions about the result are demonstrated.

6.2 Experimental Procedure

The dataset was recorded using the EMOTIV EPOC+ headset with the sampling rate of 128Sps. Real right/left finger movements were performed by five participants. Prior to data acquisition, the participants were informed of the various details of the experiments. Then, the subjects were required to fill and sign a consent form.

The participants closed their eyes and sat in a comfortable chair wearing the headset. The data was acquired for the training set and for the testing individually and the subject knows which hand to move in each session. In the training dataset, audio stimuli were given to each subject to start or stop moving their fingers. The session starts with the resting beep and after a while, a different beep occurs for informing the subject to move their fingers and then the same sequence is repeated. This procedure was repeated ten times in each session and separated with resting periods. The movement duration was six seconds while the rest periods in between had two second duration. To enlarge the number of the movement trials, overlap window of 96 sample was applied and each trial was segmented to one second so the total size of the training set is 5 subject x 2 task x 237 trial x 14 channel x 128 sample. In the testing set, only two auditory stimuli are needed to announce the beginning and the ending of the session. The subjects only move their fingers without resting periods. The duration of the testing session is 20sec. to acquire each hand finger movements. The same overlapping window and segmentation operation were applied to the testing set so its size became 5 subject x 2 task x 77 trial x 14 channel x 128 sample. Figure 6.1 illustrates the fingers movement.



Figure 6.1: Finger Movement Procedure.

6.3 Constructing BCI platform

This section describes the procedure used to construct online BCI platform based on EMOTIV EPOC+ headset. Since the EMOTIV EPOC+ is used in this thesis and the company providing this headset doesn't allow direct access to the headset without using their software and license. For this reason, we have to follow the EMOTIV instructions to establish a direct link to the EMOTIV headset.

EMOTIV provides software written in different programming languages such as C, C++, python ...etc. This software has the ability to acquire the EEG raw data directly and store it into CSV files. Our offline studies are implemented using MATLAB but unfortunately, MATLAB doesn't support multiprocessing facility which is crucial for

real time processing. For these reasons, our thesis has to use a programming language supported by the EMOTIV company for acquiring EEG data and at the same time makes use of the MATLAB facilities. C# was chosen due to having both abilities.

The online EEG signals are processed and classified using the enhanced method (described in section 5.5) which is programmed using MATLAB. This algorithm was converted to a DLL package to avoid reprogram it using another programming language. MATLAB provides library compiler in the application gallery to generate the MATLAB DLL file. It is usually recommended not to use too complex input/output parameter variables such as structure variables in these DLL files. The scalar variables (integer, double ... etc.) and no more than two-dimensional arrays are recommended to use as parameters for the MATLAB DLL.

6.4 EMOTIV EPOC+ Configuration

EMOTIV EPOC+ provides EEGLogger console application program for acquiring EEG data from the headset, do all cloud communications and store the data into CSV file. Real time EEG raw data acquisition is not allowed directly without installing some EMOTIV software and entering different secure passwords (EMOTIV, 2008). A brief description of this configuration is given as follows.

The first step towards EMOTIV configuration is installing Cortex which is a new and versatile application programming interface API for interacting with EMOTIV products, including the Insight, EPOC+ and EPOC+. To install Cortex, get it from your EMOTIV account. Installer includes Cortex UI application and Cortex Service. It provides some services like electrode calibration, EMOTIV headset connection ...etc. These facilities are provided to the users after entering their EMOTIV account and password. The user can make use of Cortex to create some applications, games, record data for experiments, and more. The Cortex protocol consists of three building standard blocks: WebSockets, JSON, and JSON-RPC. WebSockets provides a real-time connection to the underlying Cortex service, designed to be easy to use in both desktop and web-based applications. Cortex uses a widely supported format provided by JSON to send and receive data. JSON-RPC is a standard way of using JSON to make requests and get back the results (EMOTIV, 2008). Figure 6.2 shows the platform of Cortex UI.

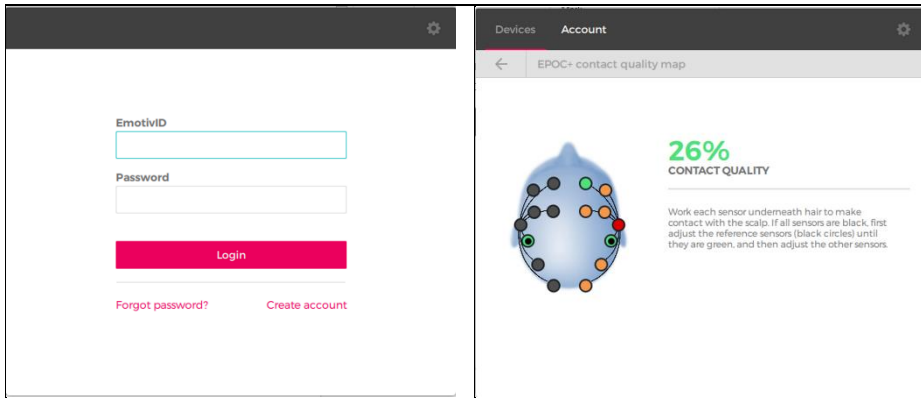


Figure 6.2 : EMOTIV Cortex UI Platform.

Secondly, getting EMOTIV authorization with a client ID and secret. These two security keys are requested from the EMOTIV account. Then, client ID and secret are typed in C# class called *accesscontroller*. Also, you have to purchase an EMOTIV professional license and type it in *EEGLogger* class with your account name and password. Now, the program is configured and ready to acquire EEG raw data directly from the EMOTIV headset.

6.5 GUI for Online BCI

The real time software of this thesis consists of different panels. Figure 6.3 shows the platform of this software.

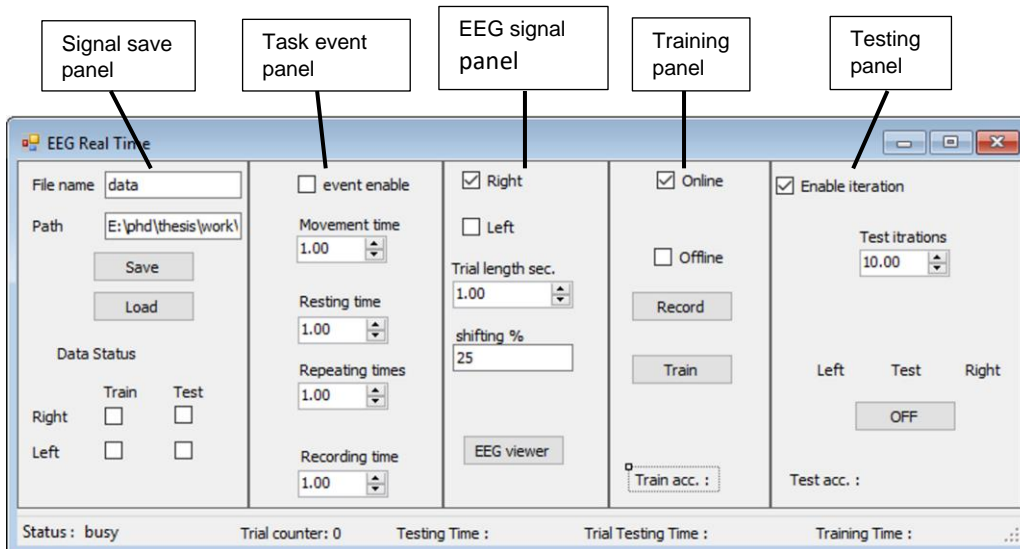


Figure 6.3: Real Time Software Platform And Its Panels.

6.5.1 Signal save panel

This panel provides a facility to save the acquired EEG signal into a specified location. It also indicates which task and dataset were acquired. It has the ability to load the previously saved EEG signal for simulation and retrieves its previous signal and event setting. It also saves metadata of the processed EEG signal like testing time, training time...etc. Saving and loading the EEG raw data are done using the MATLAB workspace file to provide further offline analysis using the MATLAB facilities.

6.5.2 Task event panel

Synchronization between EEG recording and the actual action performance plays an important role in EEG classification. This panel provides this facility by generating auditory stimuli. It has two different modes of generating the auditory stimuli to inform the subject about the action timing. The first mode is the normal mode which generates two different stimuli (starting stimulus and ending stimulus) to record unique action (e.g., right hand movement). The second mode is timing mode which generates three different stimuli to record two different actions (movement and resting actions). The user can switch between the timing/normal modes by checking or unchecking the *event enable* checkbox respectively. This panel approximately provides the events to the subject because it depends on the EMOTIV API (EEGLogger) which deal with the internet.

6.5.3 EEG signal panel

In this panel, the user can set up the trial length, the movement (right and left), the shifting percentage (percentage of displacement), and view the recorded signal. The Training and the Testing panels make use of the trial length to segment the recorded data for constructing the trials of both datasets. The (right or left) determines the task that will be recorded or tested. The shifting percentage specify the amount of the overlapping between the successive EEG trials. The following equation specifies the amount of the overlapping percentage:

$$\text{Overlapping percentage} = 100 - \text{shifting percentage} \quad (1)$$

This panel provides the ability to view the online EEG signal by pressing the *EEG viewer* button. Figure 6.4 shows the EEG viewer window.

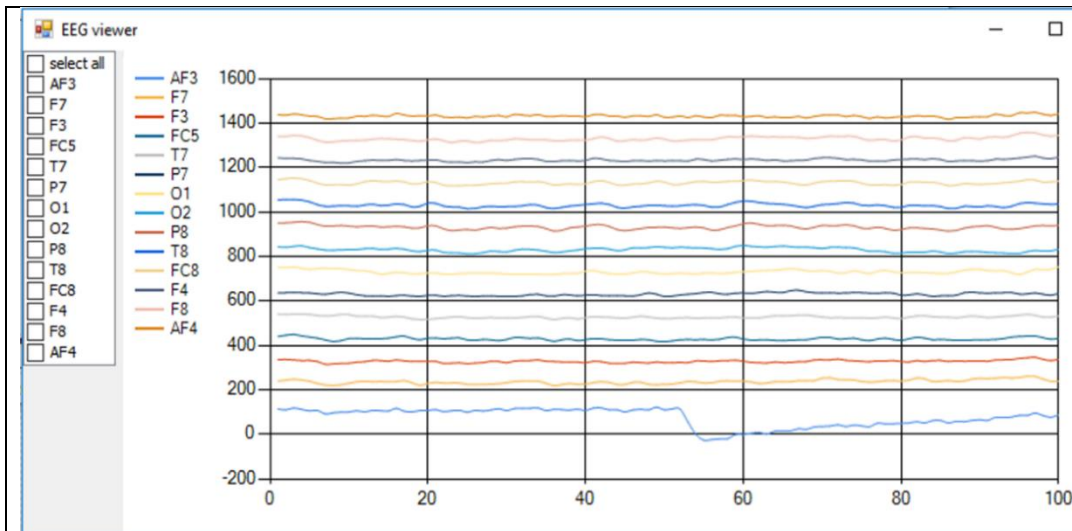


Figure 6.4: EEG Viewer Window.

6.5.4 Recording and training panel

This panel has the ability to record EEG raw data. Pressing the *record* button will initiate three different procedures: acquiring EEG raw data, generating the auditory stimuli, and storing the acquired EEG signal. The first procedure will call the modified version of the EMOTIV *EEGLogger* class which is responsible for acquiring EEG raw data from the EMOTIV headset to the computer. The *EEGLogger* class was modified to acquire EEG data and store it into a C# variable instead of a CSV file. It also modified to acquire only 14 EEG channels (AF3-AF4) because the EMOTIV EPOC+ had been providing EEG raw data of 39 samples and each sample has 16 bits. The following shows the details of each provided EEG raw package:

```
COUNTER INTERPOLATED AF3 F7 F3 FC5 T7 P7 O1 O2 P8 T8 FC6 F4 F8 AF4
RAW_CQ GYROX GYROZ MARKER SYNC TIME_STAMP_s TIME_STAMP_ms
CQ_AF3 CQ_F7 CQ_F3 CQ_FC5 CQ_T7 CQ_P7 CQ_O1 CQ_O2 CQ_P8 CQ_T8
CQ_FC6 CQ_F4 CQ_F8 CQ_AF4 CQ_CMS CQ_DRL (EMOTIV, 2008)
```

The second procedure is to generate the auditory stimuli according to the setting of the Task event panel. The timing of the composition of the auditory stimuli depends on the reception of the data from *EEGLogger* class which in turn depends on the internet connection (EMOTIV server). As we know, the speed of the internet is dynamic so an approximate timing of the auditory stimuli is provided to the subjects. This procedure

also creates an event variable whenever *event enable checkbox* checked to distinguish between the performed tasks (movement and rest). The last procedure assigns the acquired EEG data to the appropriate C# variable according to the setting of the EEG signal panel.

This panel also provides the ability to train the SVM classifier with acquired EEG signals. The training phase starts by pressing on the *train* button which calls the MATLAB DLL. The MATLAB DLL extracts the cognitive signal from the acquired EEG signal and then segments it into trials according to the setting of the EEG signal panel (*Trial length sec.*). The MATLAB DLL also uses the shifting percentage (*shifting %*) to enlarge the training set by overlapping the EEG trials according to equation 1. After the training is completed, the training accuracy will be shown beside the label *Train acc.*

6.5.5 Testing panel

This panel presents an interactive online prediction to the acquired EEG raw signal. Since, the testing phase of the SVM manipulates only one EEG trial at a time so the software can show the instantaneous prediction of each EEG trial by highlighting the labels (Right and Left), see Figure 6.5

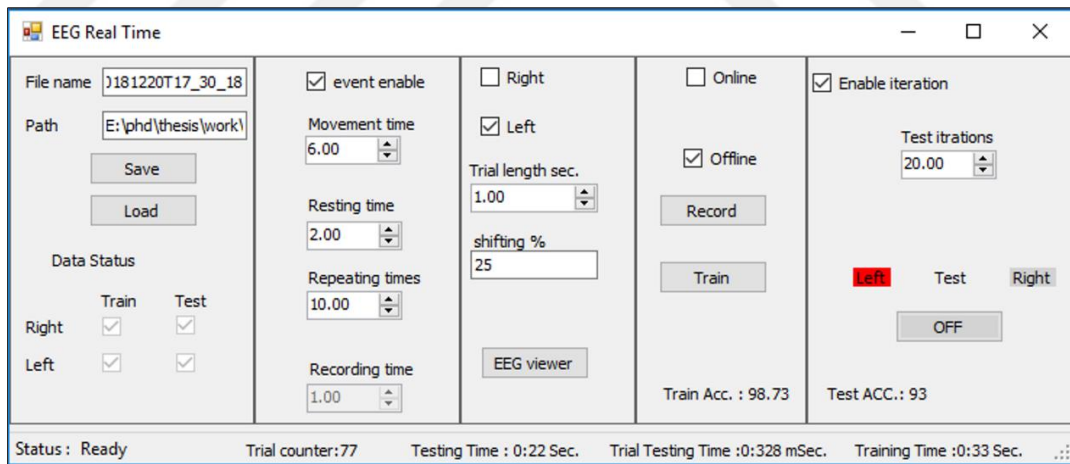


Figure 6.5: Thesis Software on Action.

The status bar of the BCI software reveals the training time, the testing time of the current trial and the total time of testing all trials. Also, all information (testing time, trials testing times, training set ...etc.) are saved into the MATLAB workspace file during saving procedure. The tested trials have to have the similar length of the trained trials so this panel uses the setting of the EEG signal panel. The user only sets the number of the tested trials in the *Test iterations*.

Pipeline processing was simulated in this panel which is one kind of parallel processing. This simulation decreases the response time of the system. The pipeline consists of two stages. Figure 4.4 illustrates the structure of the simulated pipeline.

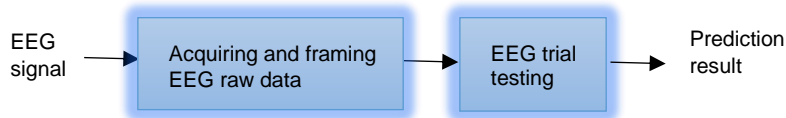


Figure 6.6: The Pipeline Structure.

The first pipeline stage is responsible for acquiring EEG raw data, storing the acquired data into FIFO buffer, and constructing EEG trials. The constructed trials are formed according to the setting of the EEG signal panel. It also uses the shifting percentage to enlarge the number of the tested trials. The second stage performs the testing phase using the MATLAB DLL. The solution of the concurrent processing problem named producer/consumer was used to simulate the behavior of the pipeline by considering the first stage as a producer and the second stage as a consumer. The solution of the producer/consumer was mentioned in chapter 4.

6.6 Processing time

In the training phase, the user presses the *train* button to start training the SVM classifier. The real-time BCI software (C# windows application program) forwards the recorded EEG raw data with the event variable to the MATLAB DLL for training. The DLL program constructs the EEG motor epochs based on the event variable and the specified configurations. According to our configuration (1sec trial length, 25% displacement), The DLL constructed 237 training trials for each EEG task. Then, the DLL program starts training the SVM classifier and returns back the trained configuration with the training accuracy.

In the testing phase, this section saves the processing times elapsed to test the EEG raw data to determine which pipeline stage is the bottleneck stage. These measurements were taken to test the 20sec of right/left finger movement of five subjects. The EEG trials were constructed to be a 1sec trial with a displacement of 0.25sec (75% overlap window). This produced 77 overlapped trials.

The most important thing for constructing the pipeline is to determine the number of stages in the pipeline and to determine the bottleneck stage (the stage that has the

largest delay time). Figure 6.7 shows the processing time of the second stage for the pipeline (EEG trial testing stage). This figure illustrates the processing time of 77 trials of 5 subjects. The pipeline total processing time of all 77 trials are 39sec for every 5 subjects. This means that the acquiring and classification of each EEG trial took about 500milisecond and this time represents the minimum time required to trigger the pipeline stages. Since the second stage took only about 100milisec to process each EEG trial then the first pipeline stage is the bottleneck of the pipeline.

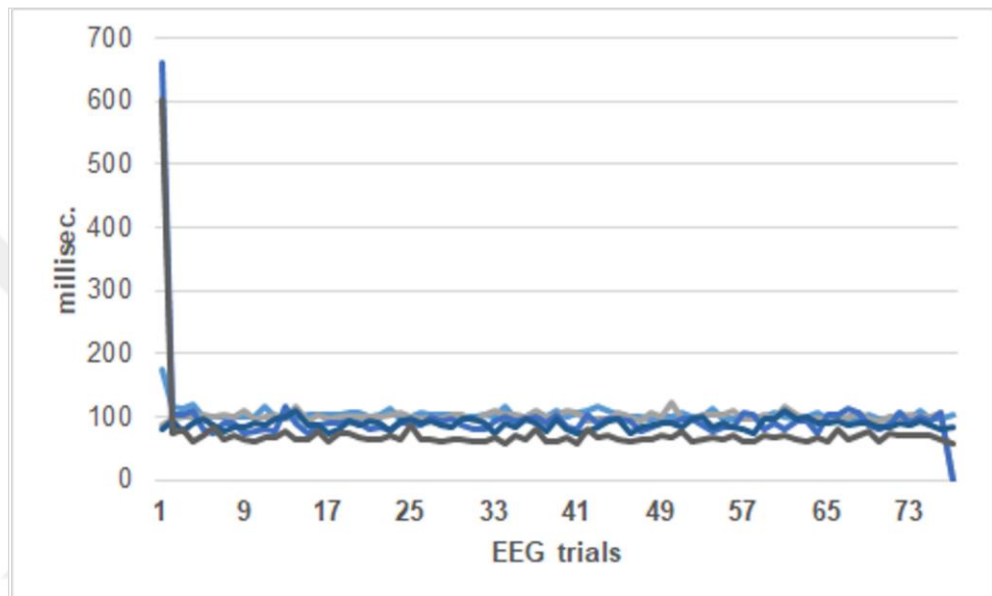


Figure 6.7: The Processing Time of The Pipeline Stage Two.

6.7 Classification rates

The classification rates of the five subject are shown in Figure 6.8. This figure shows the accuracies of right/left fingers movements and also the overall accuracy. It is observed that the overall accuracy varies between 60% to 82% with average accuracy of 68%. Table 6.1 states a performance comparison with other works. It is obviously from the table, classification performance of the online systems are relatively lower than the offline studies.

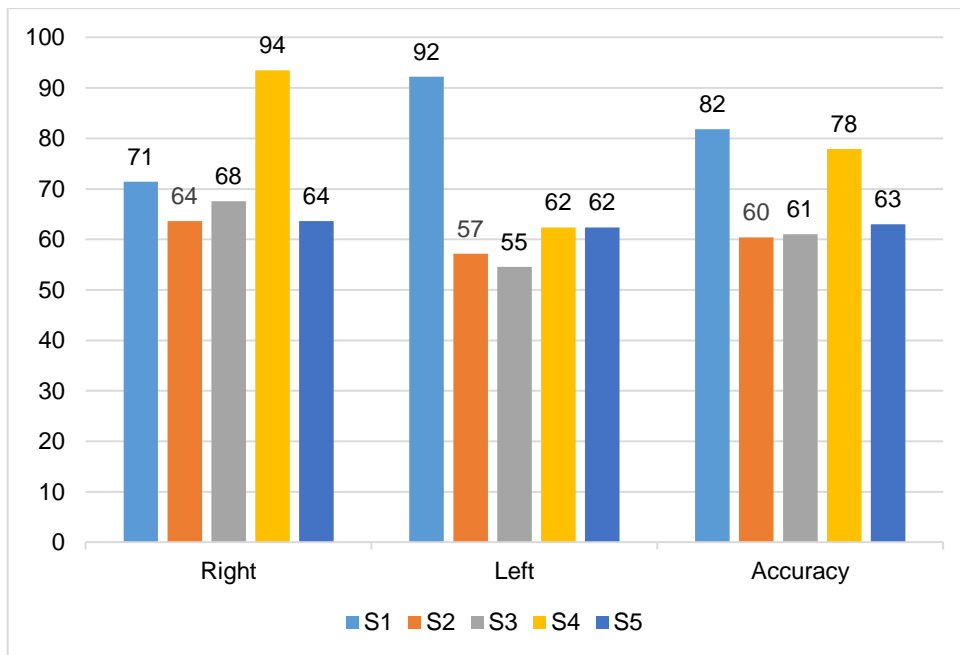


Figure 6.8: Real Time Classification Rates of Five Subjects (S1-S5).

Table 6.1: Performance Comparison with Other Studies

Method	Device	Task	Number of classes	Online testing trials	Accuracy
Enhanced method	EMOTIV EPOC+	Real right/left finger movement	2	77	60%-82%
Filtering, Interval type-2 fuzzy logic based fusion- ANFIS (Bhattacharyya, Basu, & Amit Konara, 2015)	EMOTIV EPOC+	Control robot arm	5	20	65%-70%
Filtering, SVM (Risangtuni, Suprijanto, & Widoyatriatmo, 2012)	EMOTIV EPOC+	Imaginary Right/left hand movement	2	1	60%-45%
Thresholding and rhythm extraction using FFT (Mahajan & Bansal, 2017)	EMOTIV EPOC+	Eye blink	1	0	-
Elpha and beta rhythm, CSP,LDA (Belwafi, Romain, Ghaffari, Djemal, & Ouni, 2018)	OpenBCI	Right/Left hand movement	2	120	75%

6.8 Discussions

The real time classification is a challenging task because it suffers from synchronization issues and it deals with an extremely dynamic environment which is affected by noise and artifacts. The results show that the processing time (classification time) of each EEG trial took about 100 ms except for the first trial. This exception is caused by bringing the software from computer RAM to its cache since a simulation software is used. In nonpipelined system, this 100 msec. delay time will be accumulated to the system response time proportionally with the number of EEG trials. This means the gap between the actual movement and the system response will grow up during system operation. For example, the duration between the system response and the actual movement for the 20th EEG trial will be 2 secs and this will be getting worse whenever the amount of EEG trials are increased. Figure 6.9 illustrates the response time of nonpipelined and pipelined system for only 20 trials. As shown in the figure, the pipelined system has almost constant response time. The 600 msec. of the pipeline represents the processing time of its two stages. Comparing the delay times of the pipeline stages leads to the conclusion that the bottleneck stage is the acquisition stage.

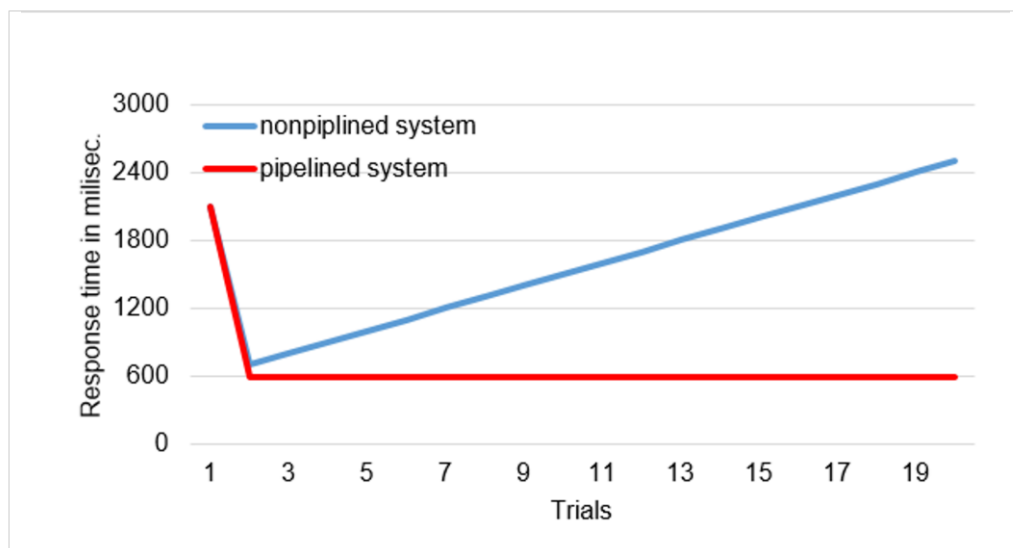


Figure 6.9: Response Time of Nonpipelined And Pipelined System.

The shifting percentage changes the number of processed trials that are constructed from the acquired EEG signal. Table 6.2 lists the number of the obtained trials from

different shifting percentages. Decreasing the shifting percentage will increase the number of trials which leads to increase the processing time and vice versa. However, using pipeline system makes this fact not true because it depends on parallel processing fashion. Therefore, it can process three different sizes of the acquired data (20, 39 and 77 trials) at a similar time which is approximately equal to the time for data acquiring (20 sec.), see Figure 6.10. This figure illustrates the theoretical, and simulation processing times with different shifting percentage. It shows that there is a small difference between the theoretical and simulation processing time. This difference is caused by the unideal sampling rate (128 samples per second) provided by C# timer. The real processing times have values between (20-39 Secs). This variation between the theoretical and real processing time is caused by the dynamic speed of internet required for handshaking with the EMOTIV server.

Table 6.2: Shifting Percentage And Its Number Of Trials

Shifting percentage	Overlapping Percentage	Number of trials
100	0	20
50	50	39
25	75	77

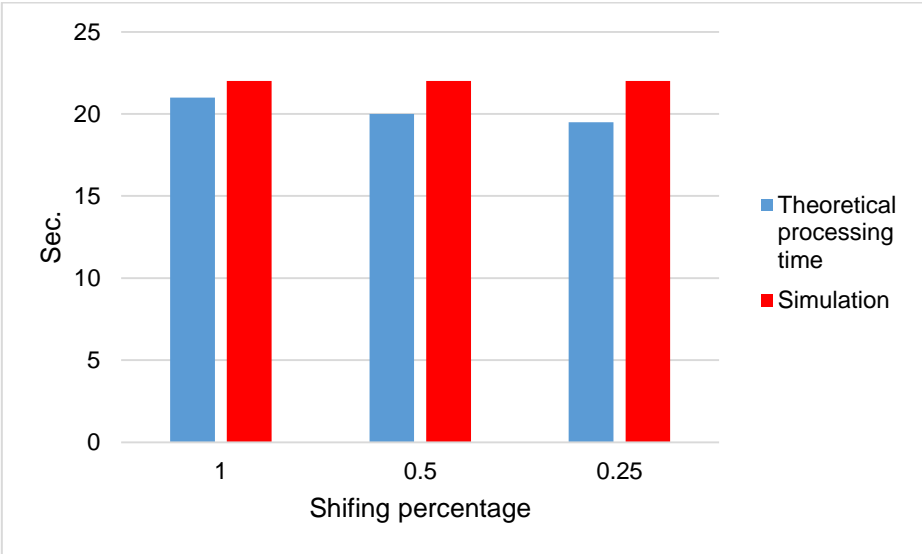


Figure 6.10: Theoretical And Simulation Processing Time of Different Shifting Percentage.

In (Al-Dabag & Ozkurt, June 2018) the average of 10-folds cross-validation classification rate of SVM classifier for 13 subjects is 98% which, implements the same feature extraction and preprocessing method but it was an offline study. Similar cognitive EEG tasks were classified in this study but the testing duration of each task was 20 second. The classification rates of this study has performance degradation compared with the results of offline classification. However, there were two subjects that have classification accuracies (82-78) and the others have accuracies around 60s. There are a lot of reasons for this kind of performance degradation. One of them, the offline classification performs cross-validation method to evaluate the classifier performance unlike online study which uses the holdout method. The subject also get tired even after only 1 minute of acquiring EEG signals unlike offline classification which perform only 20sec. of EEG recording. This kind of pain signals are also add noise to the noisy signal. Bad electrode connections also add different Gaussian noise ratio in each EEG channel. These dynamic noise ratios are also changed in each recording session.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Summary

The main objective of this study is to design and implement an adaptive wireless brain computer interface system for classifying hand movements using EMOTIV EPOC+ headset. The study includes two main parts; offline and online studies. In the offline study, an effective method for classifying EEG signals based on the cross-correlation was proposed. The performance of the algorithm was tested both with a benchmark data and a newly recorded dataset (Al- Dabag & Ozkurt, 24 October 2018). Further modification of the algorithm was implemented by proposing an artefact removal algorithm in (Al-Dabag & Ozkurt, June 2018). Also, the best feature set was selected using a genetic algorithm (Al -Dabag, Ozkurt, & Al-Aimam, 4-6 October 2018).

In the second part of the study, an online BCI system was constructed to handle the real time EEG data acquisition issues based on the detection algorithm proposed in the first part.

7.2 Conclusions

The conclusions illustrating the challenges and proposing the solutions will be given in this section:

1. In the offline algorithm, to obtain more discriminative features, the left and right hemisphere channels are cross-correlated and statistical parameters of those vectors are calculated. As mentioned in Chapter 5, this method provides a better classification performance. The first offline algorithm uses an artifact removal method based on baseline subtraction. However, the baseline (resting periods) may suffer from noise corruption. This operation may add some noise to the motor signal rather than denoising and it may lower the classification performances. For this reason, some modifications have been made to the proposed method for overcoming this problem.
2. Denoising the EEG signal plays an important role for having better classification rates. The enhanced method has better classification rates for both classifiers (SVM

and MLP) using the offline dataset, see Figures (5.19 & 5.20). This method removes EOG and EMG artifacts from the EEG motor signal using DWT and thresholding techniques.

3. The experiments show that the examined classifiers (MLP and SVM) had competitive classification rates but MLP needs some experiments to find the number of the nodes in the hidden layer while SVM doesn't need this additional experiments.
4. The genetic algorithm performed only eight rounds to find the best feature statistical methods. Finding the best feature statistical methods may need only limited iterations using GA because of using the non-deterministic searching algorithm.
5. The most discriminative features after feature selection provides a limited improvement in the classification rates of both classifiers see Figures (5.25 & 5.26).
6. The computation time of SVM is significantly less than the MLP and this fact is still true even after feature selection, see Figure (5.27 & 5.28). The most discriminative features after feature selection reduce the computation time for both classifiers, see Figures (5.27 & 5.28).
7. In online BCI, using two different programming languages provides a facility to make use of their abilities and construct a hybrid BCI software that mixes between these benefits (the scientific facilities of MATLAB and the parallelism of C#) of two programming languages.
8. The proposed BCI software is suitable for upgrading and improvements especially for the preprocessing and classification method. This operation will be accomplished by only replacing the MATLAB DLL file with its modified version.
9. Dealing with online BCI systems leads to conclude that these systems have to have two important factors: quick and accurate. As described in Chapter 6, both conditions are met by the proposed system.
10. Parallel processing represented by pipeline parallel processing represented by pipeline technique provides faster manipulation with the online EEG raw data unlike using traditional sequential systems see Figure (6.9). The pipeline system has the ability to process three times of overlapped EEG data at the time of acquiring the actual EEG data, see Figure (6.10). The processing time of the theoretical and the simulation are approximately compatible. The small difference between the two processing times caused by the unideal sampling rate (128 Sps) of the C# timer. The real processing time is equal or twice the processing time of the

theoretical and the simulation because the system deals with the internet to communicate with the EMOTIV server. However, it is still better than using nonpipeline system.

11. Comparing the offline and the online classification rates show that there is a degradation of the classification performance. There are many factors for this kind of degradation. One of them is the changing of the recording software from EmotivXavierPure in offline system to EEGLogger C# class in the online system. Another factor is replacing the classifier estimation method from 10-fold cross-validation used in the offline system with the holdout method used in the online case. The session duration, the bad electrode status, noise and recording environment are other factors that may leads to the performance degradation.

7.3 Future Works

Since BCI systems are expected to be commonly used in the future, the studies on this area will continue. Thus, it is possible to make some modifications to enhance the performance of the thesis.

- 1 Using a better headset to have a clear electrode connection would produce a better performance.
- 2 Dry electrodes may avoid the problem of bad connection caused by copper-oxide.
- 3 The denoising method can be enhanced to get rid of the Gaussian noise as well as the EOG and EMG noises.
- 4 Different feature extraction and classification methods can be used to improve the BCI performance.

REFERENCES

- Addison, P. S. (2017). *The Illustrated Wavelet Transform Handbook Second Edition Introductory Theory And Applications In Science, Engineering, Medicine And Finance*. CRC Press.
- Al- Dabag, M. L., & Ozkurt, N. (24 October 2018). EEG Motor Movement Classification Based On Cross-Correlation With Effective Channel. *Signal, Image and Video Processing, Springer*. doi:<https://doi.org/10.1007/s11760-018-1383-9>
- Al -Dabag, M. L., Ozkurt, N., & Al-Aimam, S. M. (4-6 October 2018). Feature Selection And Classification Of EEG Finger Movement Based On Genetic Algorithm. *2018 Innovations in Intelligent Systems and Applications Conference (ASYU)*. Adana: IEEE.
- Al-Aimama, S. M. (2013). *Independent Component Analysis of EEG Signal During Various Mental and Movement Tasks*. Mosul, Iraq: Technical College / Mosul.
- Al-Dabag, M. L., & Ozkurt, N. (June 2018). EEG Finger Movement Classification Based On Cross Correlation With Artifact Detection. *International Conference on Theoretical and Applied Computer Science and Engineering*. Istanbul.
- Alireza, G., Esmat, R., Ali, M. P., Mehdi, K., & Farhad, R. (2017). Automatic Channel Selection In EEG Signals For Classification Of Left Or Right Hand Movement In Brain Computer Interfaces Using Improvedbinary Gravitation Search Algorithm. *Biomedical Signal Processing and Control, Elsevier*, 109-118.
- Awad, M., & Khanna, R. (2015). *Efficient Learning Machines Theories, Concepts, And Applications For Engineers And System Designers*. Apress.
- Aydemir, O., & Ergün, E. (2019). A Robust And Subject-sSecific Sequential Forward Search Method For Effective Channel Selection In Brain Computer Interfaces. *Journal of Neuroscience Methods*, 60–67.
- Bahareh, N., Mohammad, N. R., D. T., & Vinod, C. (2017). Evolutionary Computation Algorithms For Feature Selection Of EEG-Based Emotion Recognition Using Mobile Sensors. *Expert Systems With Applications, Elsevier*, 143-155.

- Belwafi, K., Romain, O. ., Ghaffari, F., Djemal, R., & Ouni, B. (2018). An Embedded Implementation Based On Adaptive Filter Bank Forbrain–Computer Interface Systems. *Journal of Neuroscience Methods, Elsevier, 305*, 1–16.
- Bhattacharyya, S. t., Basu, D., & Amit Konara, T. D. (2015). Interval Type-2 Fuzzy Logic Based Multiclass ANFIS Algorithm For Real-Time EEG Based Movement Control Of A Robot Arm. *Robotics and Autonomous Systems, Elsevier, 68*, 104–115.
- Bhattacharyya, S., Sengupta, A., Chakraborti, T., Konar, A., & Tibarewala, D. N. (2013). Automatic feature selection of motor imagery EEG signals using differential evolution and learning automata. *Medical & Biological Engineering & Computing, Springer, 52*, 131–139.
- Burrus, C. S., Gopinath, R. A., & Guo, H. (1998). *Introduction To Wavelets And Wavelet Transform*. New Jersey: Printice Hall.
- Chandaka, S., Chatterjee, A., & Munshi, S. (2009). Cross-Correlation Aided Support Vector Machine Classifier For Classification Of EEG Signals. *Expert Systems with Applications, 1329–1336*.
- EMOTIV. (2008, May 3). (EMOTIV) Retrieved 12 5, 2018, from <https://emotiv.github.io/cortex-docs/#introduction>
- F.Ince, N., Goksu, F., H.Tewfik, A., & Arica, S. (2009). Adapting subject specific motor imagery EEG patterns in space–time–frequency for a brain computer interface. *Biomedical Signal Processing and Control, Elsevier, 4*, 236-246.
- Fausett, L. (1994). *Fundimental Of Neural Network Architectures, Algorithms And Applications*. Prentice-Hall, Inc.
- Foody, G. M., & Mathur, A. (2004). A Relative Evaluation of Multiclass Image Classification By Support Vector Machines. *Transactions On Geoscience And Remote Sensing, IEEE*.
- Fugal, D. L. (2009). *Conceptual Wavelets In Digital Signal Processing An In Depth Practical Approach For The Non-Mathematician*.
- Golilarz, N. A., & Demirel, H. (2017). Image De-Noiseing Using Un-Decimated Wavelet Transform (UWT) With Soft Thresholding Technique. *9th International Conference on Computational Intelligence and Communication Networks*.

- Graimann, B., Allison, B., & Pfurtscheller, G. (2010). *Brain–Computer Interface Revolutionizing Human–Computer Interaction*. Springer.
- Guzmán, A. S., Heute, U., Stephani, U., & Galka, A. (2017). Comparison Of Different Methods To Suppress Muscle Artifacts In EEG Signals. *SIViP, Springer, 11*, 761–768.
- Handy, T. C. (2009). *Brain Signal Analysis Advances in Neuroelectric and Neuromagnetic Methods*. Massachusetts Institute of Technology.
- Huan, L., & Hiroshi, M. (2008). *Computational Methods of Feature Selection*. Taylor & Francis Group, LLC.
- Ingle, V. K., & Proakis, J. G. (2012). *Digital Signal Processing Using Matlab 3rd edition*. Global Engineering.
- Izabela, R. (2014). Genetic Algorithm With Aggressive Mutation For Feature Selection In BCI Feature Space. *Formal Pattern Analysis & Applications, Springer*.
- J. R. Wolpaw, D. J. McFarland, & T. M. Vaughan. (2000, June). Brain–Computer Interface Research at The Wadsworth Center. *Transactions On Rehabilitation Engineering, IEEE, 8*.
- Jafarifarmand, A., Badamchizadeh, M. A., Khanmohammadi, S., Nazari, M. A., & Tazehkand, B. M. (2018). A New Self-Regulated Neuro-Fuzzy Framework For Classification of EEG Signals in Motor Imagery BCI. *IEEE, Transactions on Fuzzy Systems, 1485-1497*.
- Jiawei, H., & Micheline, K. (2006). *Data Mining: Concepts and Techniques 2nd edition*. Elsevier Inc.
- John, L. H., & David, A. P. (2012). *Computer Architecture A Quantitative Approach Fifth Edition*. Elsevier.
- Junfeng, G., Pan, L., Yong, Y., Pei, W., & Chongxun, Z. (2010). Real-Time Removal Of Ocular Artifacts From EEG Based On Independent Component Analysis And Manifold Learning. *Neural Comput & Applic, Springer, 19*, 1217–1226.
- Kai, H., & Faye, A. B. (1984). *Computer Architecture And Parallel Processing*. McGraw-Hill.
- Liu, C., Fu, Y., Yang, J., Xiong, X., Sun, H., & Yu, Z. (2017). Discrimination of Motor Imagery Pattern by Electroencephalogram Phase Synchronization Combined with Frequency Band Energy. *Journal of Automatica Sinica, IEEE*.

- Mahajan, R., & Bansal, D. (2017). Real Time EEG Based Cognitive Brain Computer Interface For Control Applications Via Arduino Interfacing. *International Conference on Advances in Computing & Communications, 115*, 812–820.
- Manish, N. T., Rohan, R. F., M., M., & Ajoy, K. R. (2017). Classification Of Artifactual EEG Signal And Detection Of Multiple Eye Movement Artifact Zones Using Novel Time-Amplitude Algorithm. *SIViP, Springer, 11*, 333–340.
- Maswanganyi, C., Tu, C., Owolawi, P., & Du, S. (2018). Overview of Artifacts Detection and Elimination Methods for BCI Using EEG. *3rd International Conference on Image, Vision and Computing, IEEE*, 832 - 836 .
- McCrimmon, C. M., Fu, J. L., Wang, M., Lopes, L. S., T.Wang, P., Karimi-Bidhendi, A., . . . Nenadic, Z. (2017). Performance Assessment of a Custom, Portable, and Low-Cost Brain-Computer Interface Platform. *Transactions on Biomedical Engineering, IEEE*.
- Miao, M., Zeng, H., Wang, A., Zhao, C., & Liu, F. (2017). Discriminative spatial-frequency-temporal feature extraction and classification of motor imagery EEG: An sparse regression. *Journal of Neuroscience Methods, Elsevier*, 13-24.
- Michael, N. (2005). *Artificial Intelligence A Guide to Intelligent Systems*. addison-Wesley.
- Misiti, M., Misiti, Y., Oppenheim, G., & Poggi, J.-M. (2007). *Wavelets And Their Applications*. Iste Ltd.
- Muhammad, Z. B., Nauman, A., Hubert, P., & Shum, L. Z. (2017). Differential Evolution Algorithm As A Tool For Optimal Feature Subset Selection In Motor Imagery EEG. *Expert Systems With Applications, Elsevier*, 184-195.
- Müller, K.-R., & Blankertz, B. (2004, Jun 14). *BCI Competition III Data Set IVa*. Retrieved from http://www.bbc.de/competition/iii/desc_IVa.html
- Nam, C. S., Nijholt, A., & Lotte, F. (2018). *Brain–Computer Interfaces Handbook Technological and Theoretical Advances*. Taylor & Francis Group.
- Norris, M., & Siegfried, D. R. (2011). *Anatomy & Physiology For Dummies® 2nd Edition*. Indiana: Wiley Publishing, Inc.
- Olkkonen, H. (2011). *Discrete Wavelet Transforms - Biomedical Applications* . InTech .

- Proakis, J. G. (2006). *Digital Signal Processing, 4th Edition*. New Jersey: Prentice Hall.
- Rakendu, R., & Reza, D. (2005). A Comparison of EEG Preprocessing Methods Using Time Delay Neural Networks. *Conference Proceedings. 2nd International IEEE Embs Conference on Neural Engineering*, 262 - 264.
- Rao, K., & Yip, P. (2001). *The Transform And Data Compression Handbook*. CRC Press.
- Rimita, L., Pratyusha, R., & Amit, K. (2017). Evolutionary Perspective For Optimal Selection Of EEG Electrodes And Features. *Biomedical Signal Processing and Control, Elsevier*, 113-137.
- Risangtuni, A. G., Suprijanto, & Widyotriatmo, A. (2012). Towards Online Application of Wireless EEG-Based Open Platform Brain Computer Interface. *Conference on Control, Systems and Industrial Informatics, IEEE*, 23-26.
- Sanei, S., & Chambers, J. (2007). *EEG Signal Processing*. England: John Wiley & Sons Ltd.
- Selim, S., Tantawi, M. M., Shedeed, H. A., & Badr, A. (2018). A CSP\AM-BA-SVM Approach For Motor Imagery BCI System. *IEEE Access*, 49192 - 49208.
- Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating System Concepts*. John Wiley & Sons.
- Silva, L., Bellon, O. R., & Boyer, K. L. (2005). *Robust Range Image Registration Using Genetic Algorithms And The Surface Interpenetration Measure*. World Scientific.
- Sim, K. G., Hussein, A. A., Kay Chen, T., Abdullah, A.-M., Chuanchu, W., & Cuntai, G. (August 2017). Automatic EEG Artifact Removal Techniques by Detecting Influential Independent Components. *IEEE Transactions On Emerging Topics In Computational Intelligence*, 1, 270-279.
- Simon Haykin. (2009). *Neural Networks and Learning Machines Third Edition*. Prentice Hall.
- Siuly, S., & Li, Y. (2012). Improving the separability of motor imagery EEG signals using a cross correlation-based least square support vector machine for brain-computer interface. *Transactions on Neural Systems and Rehabilitation Engineering, IEEE*, 20, 526 - 538.

- Siuly, W. H., & Zhang, Y. (2016). Detection of motor imagery EEG signals employing Naïve Bayes based learning process. *Measurement*, 86.
- Stephen, D. F. (2007). *Automated Neonatal Seizure Detection*. Doctor of Philosophy in Electrical and Electronic Engineering, National University of Ireland.
- Talbi, E.-G. (2009). *Metaheuristics From Design To Implementation*. John Wiley & Sons.
- Tan, P. N., Streinbach, M., & Vipin, K. (2005). *Introduction To Data Mining*. New York: Pearson Addison Wesley.
- Tim, R. M., C. A., Yu, M. C., Alejandro, O., Trevor, K., Scott, M., . . . Gert, C. (November 2015). Real-Time Neuroimaging and Cognitive Monitoring Using Wearable Dry EEG. *IEEE Transactions on Biomedical Engineering*, 62, 2553-2567.
- Wang, P., Jiang, A., Liu, X., Shang, J., & Zhang, L. (2018). LSTM-Based EEG Classification in Motor Imagery Tasks. *IEEE, Transactions On Neural Systems And Rehabilitation Engineering*, 2086-2095.
- William, S. (2012). *Operating Systems Internals And Design Principles 7th Edition*. Prentice Hall.
- Yang, K., Deng, C.-X., Chen, Y., & Xu, L.-X. (2014). The De-Noiseing Method Of Threshold Function Based On Wavelet. *International Conference on Wavelet Analysis and Pattern Recognition, IEEE*, 13-16.
- Yang, Z., Jingjun, Z., & Dongwen, Z. (2009). Implementing and Testing Producer-Consumer Problem Using Aspect-Oriented Programming. *IEEE Conferences*, 2, 749 - 752 .
- Yonghui Fang, Minyou Chen, & Xufei Zheng. (2015). Extracting Features From Phase Space Of EEG Signals In Brain-Computer Interfaces. *Neurocomputing*, 1477-1485.
- Zarei, R., He, J., Siuly, S., & Zhang, Y. (2017). A PCA aided cross-covariance scheme for discriminative feature extraction from EEG signals. *Computer Methods and Programs in Biomedicine, Elsevier*, 146, 47-57.
- Zhu, X., Li, P., Li, C., Yao, D., Zhang, R., & Xu, P. (2019). Separated Channel Convolutional Neural Network To Realize The training Free Motor Imagery BCI Systems. *Biomedical Signal Processing and Control*, 396-403.