TOWARDS EFFECTIVE APPLICATION OF DATA-DRIVEN LEARNING MODELS FOR ASSISTIVE TECHNOLOGIES AND BRAIN-COMPUTER INTERFACES

SOROOSH SHAHTALEBI

A PHD THESIS
IN
THE DEPARTMENT
OF
CONCORDIA INSTITUTE FOR INFORMATION SYSTEMS ENGINEERING (CIISE)

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY (INFORMATION AND SYSTEMS ENGINEERING)
CONCORDIA UNIVERSITY
MONTRÉAL, QUÉBEC, CANADA

DECEMBER 2020

© SOROOSH SHAHTALEBI, 2020

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: **Mr. Soroosh Shahtalebi**
Entitled: **Towards Effective Application of Data-driven Learning Models for Assistive Technologies and Brain-computer Interfaces**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Information and Systems Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

______________________ Chair
Dr. Anjan Bhowmick

______________________ External Examiner
Dr. Jane Wang

______________________ External to Program
Dr. Amir Asif

______________________ Examiner
Dr. Abdu Ben Hamza

______________________ Examiner
Dr. Nizar Bouguila

______________________ Thesis Co-Supervisor
Dr. Arash Mohammadi

Approved by ______________________
Dr. Mohammad Mannan, Graduate Program Director

November 26, 2020 ______________________
Dr. Mourad Debbabi, Dean
Gina Cody School of of Engineering and Computer Science

Abstract

Towards Effective Application of Data-driven Learning Models for Assistive Technologies and Brain-computer Interfaces

Soroosh Shahtalebi, Ph.D.
Concordia University, 2020

The worldwide population of seniors (60+) is expected to increase from 962M in 2017 to 2.1B in 2050 and 3.1B in 2100. Thus, a parallel rise in a range of age-related disorders and diseases including Parkinson's Disease (PD) and Essential Tremor (ET) is expected. Pathological Hand Tremor (PHT) is a common symptom of such disorders, which severely affects patients' quality of life. There are more than 100,000 Canadians living with PD today, but that number is expected to jump over the coming decade to more than 160,000, which necessitates an urgent quest for thoughtful planning and proactive measures. The development of assistive and rehabilitation technologies is one approach that stands to help affected individuals compensate for a variety of lost functionalities, and regain their self-sufficiency. One promising direction is to introduce a new communication medium to the human brain, known as Brain-Computer Interfacing (BCI), to bypass the impaired neural pathways and provide a direct link between the brain and an Assistive Device (AD). In addition, another approach is to capitalize on the remaining functionalities of the patients and compensating for the lost capabilities or correcting the flawed ones. In this dissertation, various types of signal processing and machine learning frameworks are developed to be utilized in the above-mentioned ADs, each enhancing our understanding of the problem in hand and surpassing its counterparts in terms of estimation accuracy, classification accuracy, adaptivity, and generalizability. In particular, a Bayesian optimization framework and an innovative multiclass classification scheme are developed to enhance the classification accuracy of BCI systems. In addition, two processing frameworks based on Adaptive Kalman filtering and Recurrent neural networks are introduced, which drastically enhance the accuracy of PHT estimation from the dynamics of tremorous hands. Moreover, a comprehensive screening protocol based on Convolutional Neural Networks is proposed for the diagnosis of PD from ET, which offers the state-of-the-art accuracy compared to its counterparts. Finally, a novel objective function for deep metric learning frameworks is developed to lessen the necessity of huge datasets to train artificial neural models in the biomedical domain, and in particular for neurological disorders. Such advancements would not only minimize the caregiving burden but could also help increase the number of productive and self-sufficient years patients have before debilitating disease symptoms take over.

Acknowledgments

Foremost, I wish to extend my sincere gratitude to my advisor, Professor Arash Mohammadi, for his patience, motivation, enthusiasm, immense knowledge, and continuous support during my studies. His guidance helped me learn the alphabet of scientific research and put me on the right track to fulfill my dreams.

Besides, I would like to thank my respected Ph.D. committee, Professors Abdessamad Ben Hamza, Nizar Bouguila, Amir Asif, and Jane Z. Wang, for their constructive comments and insightful advice at different stages of my research. Also, I would like to sincerely thank my research collaborators, Professors Rajni Patel and Farokh Atashzar, for their help during my studies.

I am also thankful to my colleagues and friends for always being there for me and giving me their endless support. Specifically, I would like to thank my friends Ali Al-Dulaimi, Afshin Ebtia, Saeed Sarencheh, and my childhood friends in "Sevom-e 3", who are like a second family to me.

Finally, my special thanks go to my beloved family, who always loved me and supported me, regardless of the path I am taking. My father, Behzad Shahtalebi, my mother, Shahin Rahiminejad, my brother, Behnam, my sister-in-law, Niloufar, and my nephew, Arshan, words cannot express how much I feel blessed to have you. My heartfelt thanks to my uncle, Dr. Hossein Rahiminejad, who planted the first seeds of enthusiasm for science in my mind and always encouraged me to pursue my dreams to the highest degree.

Contents

List of Figures

List of Tables

Abbreviations

Abbreviation	**Description**
PD	Parkinson's Disease
ET	Essential Tremor
PHT	Pathological Hand Tremor
ADL	Activities of Daily Living
BCI	Brain-Computer Interface
EEG	Electroencephalogram
AD	Assistive Device
NFB	Neuro-Feedback
ASD	Autism Spectrum Disorder
ADHD	Attention Deficit Hyperactivity Disorder
LFP	Local Field Potential
DOF	Degree of Freedom
FES	Functional Electrical Stimulation
DBS	Deep Brain Stimulation
ECOC	Error Correcting Output Coding
CNN	Convolutional Neural Networks
OVA	One vs. All
OVR	One vs. Rest
OVO	One vs. One
MI	Motor Imagery
hBCI	Hybrid Brain-Computer Interface
ERD	Event-related Desynchronization
ERS	Event-related Synchronization
SMR	Sensorimotor Rhythm
NIRS	Near Infrared Spectroscopy
ECoG	Electrocorticogram
MRI	Magnetic Resonance Imaging
CSP	Common Spatial Patterns
SVM	Support Vector Machine
LDA	Linear Discriminant Analysis
k-NN	K-Nearest Neighbors
DL	Deep Learning
MLP	Multi-layer Perceptron
RT	Rest Tremor

AT	Action Tremor
BoNT-A	Botulinum Toxin type A
BMFLC	Band-limited Multiple Fourier Linear Combiner
EBMFLC	Extended-BMFLC
PET	Positron Emission Tomography
DAT	Dopamine Transporters
SPECT	Single Photon Emission Tomography
ML	Machine Learning
NCC	Nearest Centroid Classifier
RF	Random Forest
DT	Decision Tree
ANN	Artificial Neural Networks
RNN	Recurrent Neural Network
BRNN	Bidirectional RNN
GRU	Gated Recurrent Unit
DML	Deep Metric Learning
STFT	Short Time Fourier Transform
WVT	Wigner-Ville Transform
CWD	Choi-Williams Distribution
CWT	Continuous Wavelet Transform
DWT	Discrete Wavelet Transform
TECOC	Ternary-ECOC
ICA	Independent Component Analysis
PCA	Principal Component Analysis
RHS	Right-hand Side
EOG	Electrooculogram
TSSM	Tangent Space of Sub-Manifold
ReLu	Rectified Linear Unit
RCSP	Regularized Common Spatial Patterns
SPD	Symmetric Positive-definite
db	daubechies wavelet function
AHR	Augmented Haptic Rehabilitation
RTP	Real-time Tremor Prediction
HPA	Hyper Parameter Adjustment
AR	Autoregressive
MMSE	Minimum Mean-Square Estimate
FFT	Fast Fourier Transform
NRMSE	Normalized Root Mean Square Error
PSD	Power Spectral Density
sym	Symlets wavelet function
SNR	Signal to Noise Ratio
PRF	Power Ratio Factor
MSE	Mean Squared Error
ANOVA	Analysis of Variance
QDA	Quadratic Discriminant Analysis

RBF	Radial Basis Function
SVD	Singular Value Decomposition
LSQR	Least Squares Error
AB	AdaBoost Classifier
LR	Logistic Regression
NB	Naive Bayes Classifier
ROC	Receiver Operating Characteristics
AUC	Area Under the Curve
XAI	Explainable Artificial Intelligence
Grad-CAM	Gradient-weighted Class Activation Mapping

Symbols

Symbols	Description (section of first occurrence)
Bold Uppercase letter	Matrix
Bold lowercase letter	Vector
Uppercase letter	Integer
Lowercase letter	Integer
$\boldsymbol{X}$	Each trial of EEG Recordings 3.2.1
N_{Trial}	Number of EEG trials 3.2.1
N_{ch}	Number of EEG channels 3.2.1
N_t	Number of time samples 3.2.1
Ω_i	Label of each EEG trial 3.2.1
$\mathscr{C}$	A binary coding matrix 3.2.1
γ	A classifier 3.2.1
$\Lambda(.)$	A set of binary classifiers 3.2.1
Σ	The covariance matrix of EEG signals 3.2.1
$\boldsymbol{f}$	A vector of features obtained by CSP 3.2.1
$h(.)$	A bandpass filter 3.2.1
$\mathscr{X}$	Bandpass filtered EEG signals 3.2.1
$\mathbb{B}$	An optimization particle
N_p	Number of optimization particles
$D_W(.)$	Euclidean distance between two points 3.3.1
m	margin for contrastive objective function 3.3.1
κ	Kappa value 3.3.4
P_s	Probability of correct classification 3.3.4
P_r	Probability of random classification 3.3.4
$\delta_R(.)$	Riemannian distance between two points 3.4.1
β	Eigenvalue 3.4.1
$\|\|.\|\|_F$	Frobenius norm of a matrix 3.4.1
$\boldsymbol{R}_R$	Riemannian mean 3.4.1
n	Discrete time index 4.2.1
$\boldsymbol{m}$	Measurement signal of hand motion 4.2.1
$\boldsymbol{m}_i$	Involuntary component of hand motion recordings 4.2.1
$\boldsymbol{m}_v$	Voluntary component of hand motion recordings 4.2.1
ψ	Mother wavelet function 4.2.1
u	dilation of mother wavelet function 4.2.1
j	scale of mother wavelet function 4.2.1

$\boldsymbol{T}$	Discrete wavelet transform of signal 4.2.1
$\phi_{j,u}$	Father (scaling) wavelet function 4.2.1
$\boldsymbol{m}_L$	Segmented part of PHT measurement signal 4.2.1
$\boldsymbol{p}_L(f)$	Power spectrum estimation of a signal 4.2.2
f_d	Cutoff frequency between voluntary and involuntary components 4.2.2
$\boldsymbol{m}_{i_L}^{GT}(n)$	Ground truth for involuntary component 4.2.2
$\boldsymbol{A}_L(n)$	Reconstructed approximations of signal in all levels 4.2.2
$\boldsymbol{x}(k)$	State vector for KF formulation 4.2.2
$\hat{\boldsymbol{x}}(k\|k-1)$	Predicted state in KF 4.2.2
$\boldsymbol{P}(k\|k-1)$	Estimated covariance in KF 4.2.2
$\boldsymbol{r}(k)$	Innovation term in KF 4.2.2
$\boldsymbol{S}(k)$	Innovation in covariance in KF 4.2.2
$\boldsymbol{K}(k)$	Kalman gain 4.2.2
$\hat{\boldsymbol{x}}(k\|k)$	Update term in states in KF 4.2.2
$\boldsymbol{P}(k\|k)$	Covariance update in KF 4.2.2
$\boldsymbol{w}_L(n)$	Optimized weighting vector 4.2.2
$\boldsymbol{s}_v(n)$	Synthetic voluntary component of hand motion 4.2.4
$\boldsymbol{s}_w(n)$	Additive white noise to hand motion signal 4.2.4
$\boldsymbol{s}_t(n)$	Synthetic tremor signal 4.2.4
$\boldsymbol{h}(k)$	Vector of hidden features in RNN 4.3.2
$(\hat{y})(k)$	Output vector of a RNN 4.3.2
$\boldsymbol{r}$	Reset gates in GRU 4.3.2
z	Update gate in GRU 4.3.2
$\bar{g}_{ET}$	Mean Grad-CAM pixel for ET patients 5.1
$\bar{g}_{PD}$	Mean Grad-CAM pixel for PD patients 5.1
σ_{ET}^2	Variance of Grad-CAM pixel for ET patients 5.1
σ_{ET}^2	Variance of Grad-CAM pixel for PD patients 5.1
$\boldsymbol{o}$	Extracted embeddings vector 5.4.1
c	Cosine similarity between two embeddings 5.4.1

Chapter 1

Thesis Overview

1.1 Introduction

Recent studies by United Nations [2] estimate that the population of the people over 60 years old will increase from 962 millions in 2017 to 2.1 billions by 2050, and to 3.1 billion in 2100. During the twentieth century, in western countries, the population of senior people, as well as the mean life expectancy has increased and is also expected to continue this increasing trend for the future [3]. The mean life expectancy in the US at the beginning of the twentieth century was 46.6 and 49.1 for males and females, respectively. This has increased to 69.8 and 77.5 in 1980 and is expected to reach 75.0 and 83.1 in 2040, respectively for men and women [4, 5]. Therefore, age-related neurological movement disorders such as Parkinson's Disease (PD) and Essential tremor (ET) [6–9] are expected to become more prevalent as the population of seniors over the age of sixty is increasing.

Pathological Hand Tremor (PHT) is a common upper-limb motor symptom of several age-related neurological movement disorders and is described as involuntary and pseudo-rhythmic movements [10] affecting coordination, targeting, and speed of intended motions [11]. Although PHT is not life-threatening, it severely affects the quality of life for the affected individuals to perform activities of daily living (ADLs). The growing number of affected individuals calls for the development of advanced and innovative rehabilitation and assistive technologies to help the individuals take back their lost functionalities and regain their independence in performing ADLs. To this end, a potential and promising solution is to introduce a new communication medium to the individual's brain to directly interact with the outer world and perform the action intended by the user. This approach leads to the development of Brain-computer Interfaces (BCI), which have found several clinical and commercial applications. In addition, another approach is to capitalize on the remaining functionalities of the individuals and develop Assistive Devices (AD) that enable the users to make the utmost use of their current limited actions and regain their independence in performing ADLs. Although several works have attempted to further enhance either of the above approaches, the developed methodologies are still subjective, based on simplified assumptions, and provide a very limited communication bandwidth for the users, which leads to low reliability and limited application of such technologies in the real world.

To minimize the effect of the aforementioned disorders on society and control their debilitating symptoms, this thesis aims at integrating the two strategies into a single framework and designing a collaborative procedure to enhance the throughput and reliability of the overall system. The envisioned architecture, which is also referred to as Hybrid-BCI, in addition to capitalizing on the plasticity properties of the brain, which allows it to interact with new communication mediums, employs all the available functionalities and capabilities of the affected individuals to provide a higher degree of performance and reliability in assistive services. Although the two modalities can be employed collaboratively to enhance the accuracy of the decoded signals and boost the reliability and performance of the assistive device, the contributions of the thesis can also be independently employed in different domains as a stand-alone solution. The fulfillment of the objective followed in this thesis is highly tied with the successful development of signal processing and machine learning frameworks to analyze the noisy biological signals and extract meaningful information for the development of a practical Hybrid-BCI system.

To monitor the brain's activities for the BCI component, Electroencephalogram (EEG) is employed, and to take advantage of the remaining functionalities of patients' limbs and provide an alternative to the BCI system, acceleration of hand movements are employed as a measure of hand tremor. Within the BCI component, the goal is to extract informative features from the signals to accurately classify the mental states and translate them into proper commands, however, the challenge with the processing of EEG signals is the noisy and blurred image they represent from the brain's activities. On the other hand, the challenge with hand motion signals is the tremorous movements of limbs in the affected individuals, which limits our understanding of the voluntary motion of those individuals. Thus, the major contribution of the proposed project, besides its novel multi-modal nature, lies in the signal processing block, where we need to overcome critical problems such as low signal-to-noise ratio, inter-/intra-subject variability, and the highly dynamic behavior of the studied phenomena in the measured signals.

1.1.1 Assistive Technologies

As discussed previously, age-related neurological disorders and more specifically PD and ET, share PHT as their common symptom, which degenerates the mobility capabilities of the affected individuals. Mobility, which is the most studied physical ability, plays an undeniable role in individual's quality of life. Natural selection has enabled the "engine" of mobility with great robustness, redundancy, and functional reserve [12], which facilitates acquiring complex mobility patterns, even for severely impaired individuals. Thanks to the highly efficient neurological system of motor control, age-associated physiological impairments of this system are easily compensated except for the cases that impairment can no longer be compensated. Loss of mobility in seniors may be a consequence of multiple impairments in the central nervous system, muscles, joints, and energetic and sensory physiological systems.

Individuals who are experiencing severe motor disabilities can use a variety of different Assistive Devices (ADs), which provide them a means for managing their daily needs and communicating with other people. The development of such ADs relies on the residual motor or communication skills of the user, which aims at bypassing the impaired capability by employing the remaining functionalities of the limbs. The ADs range from simple switches, which are connected to a remote controller, to complex sensors connected to a computer or an eye-tracking system [13].

In the thesis, the term "assistive device" refers to a set of computational methods to analyze the hand motion in the patients with neurological disorders and separate the tremorous component of motion from the voluntary component of motion. The analyzed signals represent either the acceleration or the velocity of hand motion. The output of the processing frameworks could then be used in various types of assistive technologies including but not limited to smart spoons, robotic rehabilitation devices, tele-robotic surgery technologies, and interfaces for computers that are specialized for compensating the tremorous hand motions.

1.1.2 Brain-computer Interfaces

BCIs are developed to establish a new communication medium for the human brain to interact with the outer world, with the goal of compensating for the lost capabilities or boosting the performance of the available functionalities. The human brain, thanks to its vast and complex network of neurons, constitutes the most powerful signal processing system ever known to us in the sense that it can simultaneously analyze and fuse several streaming signals from different modalities in an adaptive and real-time fashion. Such intriguing capabilities of the brain as well as the aforementioned motivations to develop a BCI system for age-related neurological disorders have motivated an extensive amount of research on developing BCIs [14–16] to establish direct communication links for interaction with the physical world via brain signals. The BCI systems are considered as the main building block of the human-in-the-loop, cyber-physical systems [17, 18], and are becoming a key component for assistive/rehabilitative systems [19–21], and neuro-prosthesis control [22]. The plasticity properties of the brain [23] have opened up various possibilities for the BCI systems in therapeutic applications and have significantly enhanced the effectiveness of rehabilitation therapies. Nowadays, rehabilitation-based BCIs [24] are gaining more attention in different areas such as Neuro-feedback (NFB), therapy for Autism Spectrum Disorder (ASD), Attention Deficit Hyperactivity Disorder (ADHD), schizophrenia, and motor rehabilitation for post-stroke patients to name a few. Despite the promising vision that full integration of BCIs in rehabilitation systems presents, the research in this area is still in its early stages as the performance of the aforementioned artificial systems rarely matches that of the brain. This calls for an urgent quest to further improve the performance of existing BCI systems by developing advanced and improved processing solutions, which is one of the main motivations of this work.

Among several types of brain signals which are deployed to control the movement, three of them are of more interest for BCIs, including electrophysiological signals recorded over the scalp (EEG), over the cortical surface (electrocorticography), and within the brain (single-neuron action potentials(single units) and Local Field Potentials (LFPs)) [25]. The common aspect of these methods is that they measure the extracellular potentials generated by the neurons in cortical layers; however, they differ in the field distance and spatial resolution. Although the intracortical recordings of single-neuron activity represent the highest spatial resolution, on the other hand, this method is the most invasive one which requires implanting electrodes in the parenchyma.

Some research works have uncovered how the kinematic parameters of the movement are encoded in the neuronal electrical activities. Thanks to these findings, researchers have been able to develop real-time closed-loop BCI systems. Initially, these systems were tested on nonhuman primates, then electrode arrays were implanted in the severely disabled patients for multidimensional control of a computer cursor or a robotic arm. Intracortical recordings (mostly single units) facilitate achieving a high level of degree-of-freedom (DOF) on the BCI systems; however, the long-term stability of the electrodes is still a matter of concern which limits their clinical applications. On the contrary, EEG offers a non-invasive brain imaging modality and has found many applications in BCIs, including 2-D and 3-D BCI control. Since the EEG recording is performed over the extracellular field potentials over the scalp, the spatial resolution is limited (in centimeters level) and the recordings are more prone to be interfered with environmental and physiological artifacts, like electromyographic signals from cranial muscles or electrooculographic activity. Having said all these, still, the EEG is the simplest and safest brain imaging modality with the most clinical applications.

1.1.3 Functional Electrical Stimulation

Functional Electrical Stimulation (FES) refers to an assistive technology for individuals with impaired neurological circuits, where an electrical current is injected to excitable tissues in order to assist or replace the lost neural activation. In addition to therapeutic applications of electrical stimulation, there are various use cases of them in the restoration of function in chronic applications. To be more specific, the effect of therapeutic electrical stimulation remains after the therapy, and the goal is to improve tissue health or its voluntary function. In FES systems, on the other hand, the ability to function in a certain limb is provided by stimulating the muscles or neural pathways to it, and thus, the FES systems are normally worn by the user. By means of FES systems, which are also referred to as Neuroprostheses, both sensory and motor functions could be restored, where the former consists of auditory and visual neuroprostheses. To restore the motor functions through FES systems, carefully designed electrical stimuli are injected to the lower motor neurons to elicit action potentials in the innervating axons, and therefore, activating certain muscles to produce the desired movement.

Various neurological disorders including PD and ET share a common symptom referred to as pathological tremor, which significantly affects the quality of life in the affected individuals, and thus, there has been an ongoing quest to develop techniques and solutions in order to suppress or attenuate the tremor. The prime choice in controlling the debilitating effects of tremor is pharmaceutics [26], however, their effect is temporary and undesirable side effects are inevitable [27]. In severe cases that medication is not effective, an invasive approach referred to as Deep Brain Stimulation (DBS) [28] is followed, which is based on injecting electrical currents into certain areas of the human brain to suppress their activities responsible for tremorous movements. In addition, although it is also shown [29, 30] that mechanically loading the tremorous limbs through exoskeletons could be a potential solution, this technique is cumbersome for daily and home use. An alternative solution to mechanically loading the tremorous limb is to employ FES technology to suppress or attenuate the activity of the muscles involved in tremorous movements and eventually correcting

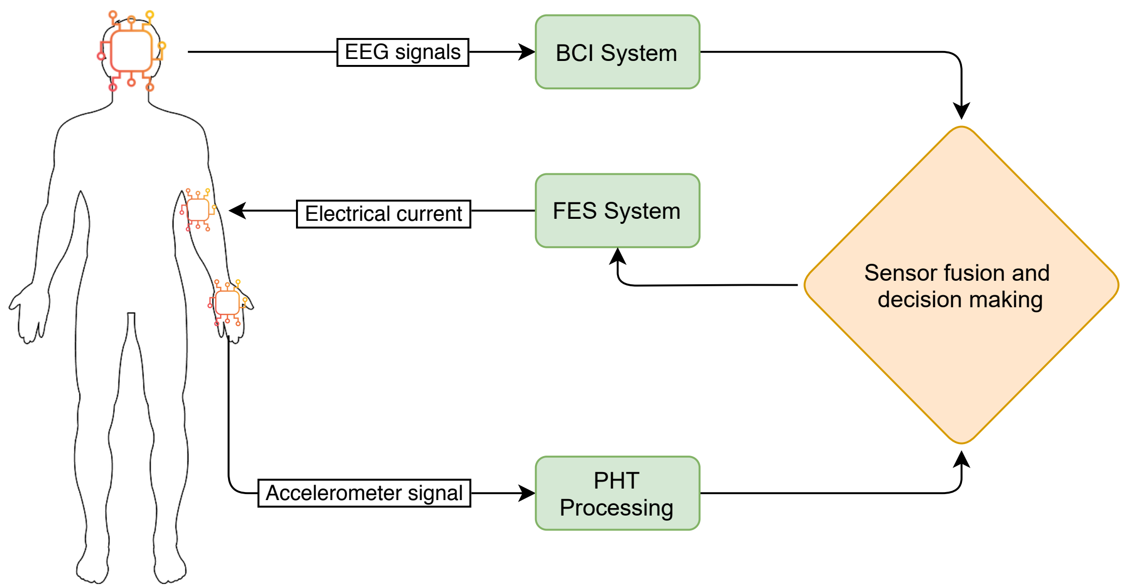


Figure 1.1: An example of tremor suppression system based on FES technology, which equipped with a BCI system and a PHT processing framework to enhance the accuracy and reliability of the system in detecting and attenuating the tremorous movements.

this abnormality. The earliest works based n this technique [31,32] employed displacement sensors to estimate the flexion or extension of the wrist joint and then injected trains of low amplitude current pulses into the muscles involved in the flexion or extension. This study showed a significant reduction in the wrist tremor while maintaining the voluntary action of the individuals.

One promising direction for the FES approach for attenuating the tremorous activities of the muscles is its integration with BCI systems, which enables the detection of voluntary (and involuntary) motor activities from the brain signals and further enhances the efficacy of the treatment. To be more specific, this approach follows the block diagram depicted in Figure 1.1, and is grounded on the following three building blocks.

- A BCI system is employed to detect the tremor onset, as well as the voluntary and involuntary motor activities in the brain.
- A processing framework is employed to process the acceleration of the hand motion and estimate the voluntary component of motion in real time and also predict the voluntary motion ahead of time.
- A multi-channel FES system is utilized for selective stimulation of muscles involved in tremorous movements, while reducing the influence on the voluntary motion of limb.

The application of FES technology for tremor suppression, as described in this section, demonstrates a real-world use case for the processing frameworks and novelties presented in this thesis. It is worth highlighting that the contributions of the thesis are widely applicable in different domains and problems, and the materials presented in this section only demonstrate one sample use case, which binds all of the contributions into a one single application.

1.2 Thesis Contributions

In this section, the research problems and the contributions of the thesis towards them are presented.

- **Chapter 3: Data-driven Methods for EEG-based Brain-Computer Interfaces [33–39]**

    1. **Development of ECCSP framework, which demonstrates the applicability of Error Correcting Output Coding (ECOC) classifiers for EEG studies [33]:** A BCI system designed to operate in real-world conditions, must be able to discriminate multiple tasks and activities. This fact expresses the urge to develop/implement classifiers intrinsically designed for multi-class problems. One such technique that is well-regarded in other fields but has not yet been applied to EEG-based classification is the ECOC. The thesis fills the mentioned gap. The BCI Competition IV-2a dataset is used to evaluate the performance of the proposed ECCSP framework. Our results showed that ECCSP achieves similar performance in comparison to the state-of-the-art algorithms but is extensively simpler with significantly less computational overhead making it a practical alternative for real-time EEG motor imagery classification tasks.
    2. **Development of ECCSP-TB framework, which introduces the Ternary-ECOC classification scheme, which boosts the classification performance in multiclass classification problems [34]:** In the thesis, a modified version of the ECOC classifiers is developed for EEG classification problems which deploys ternary class codewords. Therefore, more combinations of the classes and a greater number of classifiers vote for the final result. The proposed classifier is coupled with a Bayesian framework to compute the optimized spatio-spectral filters to extract the most discriminative feature sets of different classes. The proposed framework is applied to a motor imagery classification problem and evaluated over the BCI Competition IV-2a dataset where the results indicate a noticeable enhancement over other methods developed for multi-class EEG classification.
    3. **Development of ECCSP-TB2B framework which utilizes subject-specific optimized filter banks to extract informative features to analyze the MI signals [35, 36]:** To further individualize the spatial and spectral filters within an EEG processing framework in order to enhance the classification accuracy of the system, a novel Bayesian framework that simultaneously optimizes a number of subject-specific filter banks and spatial filters is developed. Optimized double-band spectro-spatial filters are derived based on common spatial patterns coupled with the ECOC classifiers. The proposed framework constructs optimized subject-specific spectral filters in an intuitive fashion resulting in the

creation of significantly discriminant features, which is a crucial requirement for any EEG-based BCI system.

4. **Introducing a wavelet-based dimensionality-reduction scheme for EEG processing [37]:** With the goal of optimizing the EEG-based BCI system for real-time applications and reducing the processing workload while exploiting the maximum amount of information from the EEG signals, the thesis proposes a level-based classification approach that couples the Wavelet decomposition with Riemannian manifold spatial learning (WvRiem). In the proposed WvRiem framework, the EEG signals are decomposed into several components (levels) and then spatial filtering via Riemannian manifold learning is performed on the best level which yields the most discriminating features. The proposed WvRiem is evaluated on the BCI Competition IV-2a dataset and noticeably outperforms its counterparts.

5. **Development of a deep learning framework based on Siamese neural networks for EEG processing [39]:** Despite the successful employment of deep learning methods in various domains, their application for small medical datasets always raises concerns about the curse of overfitting to the training data. The thesis addresses this unmet quest by proposing a new EEG processing and feature extraction paradigm based on Siamese neural networks, which can be conveniently merged and scaled up for multi-class problems. The idea of Siamese networks is to train a double-input neural network based on a contrastive loss-function, which provides the capability of verifying if two input EEG trials are from the same class or not. The introduced Siamese architecture, which is developed based on Convolutional Neural Networks (CNN) and provides a binary output on the similarity of two inputs, is combined with One vs. Rest (OVR) and One vs. One (OVO) techniques to scale up for multi-class problems. The efficacy of this architecture is evaluated on a 4-class Motor Imagery (MI) dataset from BCI Competition IV-2a and the results suggest a promising performance compared to its counterparts.

- **Chapter 4: Data-driven Methods for Pathological Hand Tremor Estimation [40–44]**

  1. **Development of an adaptive processing framework based on wavelet transformation and Kalman filtering to predict the voluntary component of hand motion real-time and in a myopic fashion [40, 41]:** One major bottleneck in accurately separating the voluntary and involuntary components of hand motion in patients with PHT is the highly diverse and dynamic behavioral pattern of PHT in and across patients. The thesis addresses this issue by developing a novel on-line adaptive method which can adjust the hyper-parameters of the filter to the variable characteristics of the tremor. The proposed Wavelet decomposition coupled with adaptive Kalman filtering technique for pathological tremor Extraction, referred to as the WAKE framework, is composed of a new adaptive Kalman filter and a wavelet transform core to provide an indirect prediction of the tremor, one sample ahead of time, to be used for its suppression.

The performance of WAKE is evaluated over three different datasets, where the first one is a synthetic dataset that simulates hand tremor under ten different conditions. The second and third ones are real datasets recorded from patients with PHT. The results obtained from the proposed WAKE framework demonstrate significant improvements in the estimation accuracy in comparison with two well-regarded techniques in the literature.

2. **Development of a deep learning framework based on recurrent neural networks to discriminate the voluntary and involuntary components of motion in real-time and in a myopic fashion [42, 44]:** Another major issue with estimating the voluntary and involuntary components of hand motion in patients with PHT is the unavailability of ground truth to precisely validate different processing frameworks developed for this task. The thesis addresses this unmet need by establishing a deep recurrent model to predict and eliminate the PHT component of hand motion. More specifically, we propose a machine learning-based, assumption-free, and real-time PHT elimination framework, the PHT-Net, by incorporating deep bidirectional recurrent neural networks. The PHT-Net is developed over a hand motion dataset of 81 ET and PD patients collected systematically in a movement disorders clinic over 3 years. The PHTNet is the first intelligent systems model developed on this scale for PHT elimination that maximizes the resolution of estimation and allows for the prediction of future and upcoming sub-movements.

3. **Performing a feasibility study on fusing two different datasets on hand motion recordings to train a neural network [43]:** Despite the successful employment of deep learning methods in various domains, their application for small medical datasets always raises concerns about the curse of overfitting to the training data. Since the availability of large datasets, especially in the PHT estimation field is a bottleneck, the thesis investigates the possibility of combining different recording modalities of PHT to generate a neural network for this purpose. In fact, the thesis approves the possibility of jointly using accelerometer data and gyroscope recordings to produce a larger dataset for training a relatively complex network, which can potentially be extended for a deeper generalization.

- **Chapter 5: Data-driven Methods for Discrimination of Neurological Disorders [45, 46]**

1. **Development of a multi-stage and hierarchical screening protocol and classification model for differential diagnosis of PD from ET based on the kinematics of hand motion in affected individuals [45]:** Societal aging has drastically increased the prevalence of age-related neurological disorders worldwide, such as PD and ET. PHT is a common symptom of PD and ET, which affects manual targeting, motor coordination, and movement kinetics. Effective treatment and management of the symptoms relies on the correct and in-time diagnosis of the affected individuals, where the characteristics of PHT serve as an imperative metric for this purpose. Due to the overlapping features of

the corresponding symptoms, however, a high level of expertise and specialized diagnostic methodologies are required to correctly distinguish PD from ET. The thesis proposes a data-driven model, referred to as NeurDNet, which processes the kinematics of the hand in the affected individuals and classifies the patients into PD or ET. NeurDNet is trained over 90 hours of hand motion signals consisting of 250 tremor assessments from 81 patients, recorded at the London Movement Disorders Centre, ON, Canada. The NeurDNet outperforms its state-of-the-art counterparts achieving exceptional differential diagnosis accuracy of 95.55%. In addition, using the explainability and interpretability measures for machine learning models, clinically viable and statistically significant insights on how the data-driven model discriminates between the two groups of patients are achieved.

2. **Cosine-based Objective Function for Deep Metric Learning [46]:** The unprecedented capacity of deep learning techniques in extracting high-level semantic embeddings from data has catalyzed the potent of deep metric learning for classification, verification, few-shot learning, and visual search tasks. A major bottleneck in further boosting the performance of deep metric learning is the objective function employed for recognizing the distance (similarity) of embeddings. Existing methods e.g., contrastive loss and triplet loss, often suffer from slow convergence and poor local minimum due to the utilization of only one negative instance to generalize over the pairwise distance between data points. In addition, some existing methods are solely based on the Euclidean distance between the embeddings, which makes the model significantly sensitive to the covariate shift. The thesis proposes a multi-class N-pair objective function based on cosine similarity, referred to as MPCL, where the intra-class and inter-class alignment of group embeddings are respectively maximized and minimized. Upon successful training of the network, the embeddings of each class find perpendicular directions with respect to each other, spanning in the non-negative orthant of the features space. The MPCL is thoroughly evaluated over the MNIST dataset and promising results in terms of classification accuracy and speed of convergence are obtained.

1.3 Organization of the Thesis

Chapter 1 provided an overview and a summary of important contributions made in the thesis. The rest of the thesis is organized as follows.

- Chapter 2 presents an introduction to the problem in hand, and the literature of each topic is reviewed thoroughly. Also, this chapter encapsulate the required technical background of the materials presented in the thesis.
- In Chapter 3, the contributions of the thesis in developing data-driven frameworks for EEG-based BCI systems are comprehensively presented.

- Chapter 4 revolves around the contributions of the thesis in addressing the challenges with analyzing the kinematics of hand motion in patients with neurological disorders. Various data-driven processing frameworks based on Kalman filter, wavelet transformation, recurrent neural networks are presented.
- In Chapter 5 an advanced screening protocol and data-driven framework to diagnose patients with neurological disorders based on their kinematics of hand motion is presented. This chapter also covers one of the contributions of the thesis in developing a novel objective function for deep metric learning frameworks. The introduced objective function, referred to as MPCL, is based on cosine function as the similarity metric and drastically enhances the convergence rate of its counterparts.
- Chapter 6 concludes the thesis and provides some directions for future work.

Throughout this thesis, the following notations are used unless otherwise is stated; Non-bold letter x denotes a scalar variable; Lowercase bold letter $\boldsymbol{x}$ represents a vector, and; Capital bold letter $\boldsymbol{X}$ denotes a matrix.

1.4 Publications

C-1 **S. Shahtalebi** and A. Mohammadi, "Error Correction Output Codding Coupled with the CSP for Motor Imagery BCI Systems," in *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 2071–2075.

C-2 **S. Shahtalebi** and A. Mohammadi, "Ternary ECOC Classifiers Coupled with Optimized Spatio-spectral Patterns for Multiclass Motor Imagery Classification," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2017, pp. 2231–2236.

C-3 **S. Shahtalebi**, A. Mohammadi, S. F. Atashzar, and R. V. Patel, "A Multi-rate and Auto-adjustable Wavelet Decomposition Framework for Pathological Hand Tremor Extraction," in *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2017, pp. 432–436.

C-4 **S. Shahtalebi** and A. Mohammadi, "A Bayesian Framework to Optimize Double Band Spectra Spatial Filters for Motor Imagery Classification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 871–875.

J-1 **S. Shahtalebi** and A. Mohammadi, "Bayesian Optimized Spectral Filters Coupled with Ternary ECOC for Single-trial EEG Classification," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 12, pp. 2249–2259, 2018.

C-5 **S. Shahtalebi** and A. Mohammadi, "Feature Space Reduction for Single Trial EEG Classification based on Wavelet Decomposition," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2019, pp. 7161–7164.

C-6 M. Mirgholami, S. Shahtalebi, W. Cui, R. Karimi, A. Asif, and A. Mohammadi, "Adaptive Subject-specific Bayesian Spectral Filtering for Single Trial EEG Classification," in *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP).* IEEE, 2019, pp. 1–5.

C-7 **S. Shahtalebi**, A. Asif, and A. Mohammadi, "Siamese Neural Networks for EEG-based Brain-computer Interfaces," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, 2020, pp. 442–446.

J-2 **S. Shahtalebi**, S. F. Atashzar, R. V. Patel, and A. Mohammadi, "WAKE: Wavelet Decomposition Coupled with Adaptive Kalman Filtering for Pathological Tremor Extraction," *Biomedical Signal Processing and Control*, vol. 48, pp. 179–188, 2019.

J-3 **S. Shahtalebi**, S. F. Atashzar, R. V. Patel, and A. Mohammadi, "HMFP-DBRNN: Real-time Hand Motion Filtering and Prediction via Deep Bidirectional RNN," *IEEE Robotics and Automation Letters*, 2019.

C-8 **S. Shahtalebi**, S. F. Atashzar, R. V. Patel, and A. Mohammadi, "Training of Deep Bidirectional RNNs for Hand Motion Filtering via Multimodal Data Fusion." in *GlobalSIP*, 2019, pp. 1–5.

J-4 **S. Shahtalebi**, S. F. Atashzar, O. Samotus, R. V. Patel, M. S. Jog, and A. Mohammadi, "PHTNet: Characterization and Deep Mining of Involuntary Pathological Hand Tremor using Recurrent Neural Network Models," *Scientific reports*, vol. 10, no. 1, pp. 1–19, 2020.

J-5 **S. Shahtalebi**, S. F. Atashzar, R. V. Patel, M. S. Jog, and A. Mohammadi, "NeurD-Net: An Artificial Intelligent-based Approach in Neurological Disorders Classification," *Submitted to Scientific reports*, 2020.

J-6 **S. Shahtalebi** and A. Mohammadi, "MPCL: Deep Metric Learning based on Multi-class N-pair Cosine Similarity," *To be submitted to IEEE Transactions on Neural Networks and Learning Systems*, 2020.

Chapter 2

Introduction and Literature Review

2.1 Background

In Chapter 1, the overall theme of the thesis, as well as its objectives were discussed thoroughly. In addition, a detailed list of contributions made in the thesis was presented. In this chapter, a comprehensive literature review on BCI systems, ADs, and computational techniques to differentiate PD from ET is provided. Moreover, this chapter encapsulates a summary of the required signal processing and machine learning tools that are employed in the proposed methodologies throughout the thesis.

2.1.1 Brain Computer Interfaces

The human brain is the most intriguing signal processing system in existence due to its ability to extract/fuse information from several streaming signal modalities adaptively and in real-time. In this regard, BCIs are designed via analysis of different physiological signals to establish a communication medium between the human brain and a machine [47], and have found a diverse range of applications in different fields, as shown in Figure 2.1. Although the ultimate goal of a BCI is to allow the people with movement disorders and disability to communicate in a more efficient way and despite the impressive progress in this field [48], the performance of BCIs cannot be compared with the non-BCI ADs, in terms of performance and interaction speed. To compensate for this shortcoming and deliver the expected functionality of a BCI, and also to employ the remaining functionalities of the subjects such as control possibilities [49] employment of two or more parallel interfaces, i.e., a BCI system and one or more additional communication channels is suggested which is referred to as hybrid BCI (hBCI) [50, 51]. Especially, since the physical and mental states of individuals vary over time (e.g., when the muscles are not exhausted, the muscular activity is available as a control signal, unless the brain activity could be employed), employing various control strategies ensures a robust and reliable performance for the AD. The additional communication channels could be through any physiological signal or any special input device such as joystick or switches. To be more specific, various communication channels could be employed to run different modules of an AD or all of them

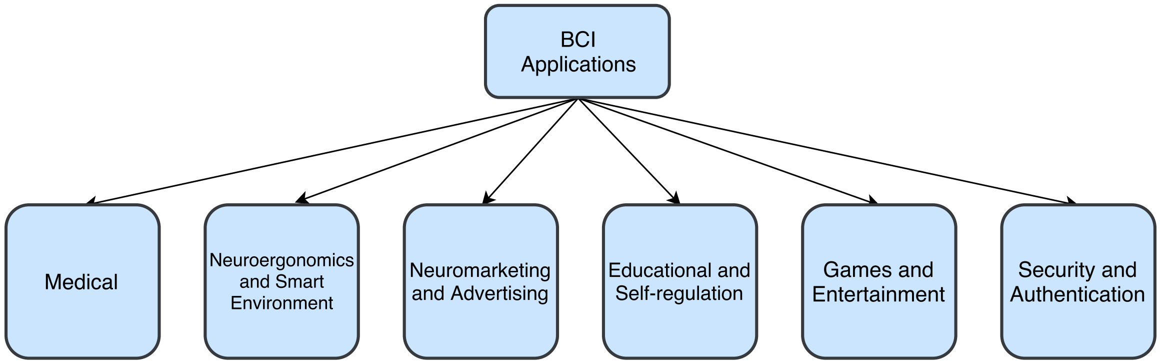


Figure 2.1: Different applications of BCI systems [1].

could be combined together [49] to allow the user to flexibly employ any of the communication channels, based on their preference and performance. The hybrid employment of the communication channels is expected to improve the quality of life of a patient.

BCI for Motor Applications

Recently, numerous studies have been conducted to develop a BCI technology to further improve the quality of life and restore the functionality to the people with severe neural injuries and motor disabilities. The application of BCI systems in rehabilitation could be categorized into two major fashions [19], which are introduced here.

- The first approach, which is simple and has been extensively studied in the literature states that the BCI systems can replace the damaged neuromuscular system. In this fashion, the brain signals are utilized to enable the patient to interact with the environment. Consequently, a person can perform a variety of tasks like answering to "yes" or "no" questions, controlling the cursor on a computer or controlling a robotic arm by using electrophysiological signals such as EEG of the brain.

- The second approach which is a more complex and novel is based on inducing activity-dependent brain plasticity to restore more normal brain function. For example, by focusing on a motor task or by requiring the activation or deactivation of specific brain signals, one could modify brain plasticity.

While the first approach is more passive, the latter one is more active and demands full participation of the patient in the process of interfacing with a computer. Several studies on different brain imaging modalities have observed that neurophysiological rhythmic activities recorded over the sensorimotor cortex are modulated by actual movement, motor intention or motor imagery [25]. The modulation is proven to be a decrease in the alpha band ($8-13$ Hz or also know as μ rhythm) and beta band ($14-26$ Hz), also known as Event-related Desynchronization (ERD), accompanied by an increase in the gamma frequency band (>30 Hz), know as Event-related Synchronization (ERS). These rhythmic activities are named as Sensorimotor Rhythms (SMRs). Motor intention or motor imagery can be decoded from SMRs which is served as the basis for SMR-based BCIs. More importantly, the ERD and ERS have found to be in a somatotopic manner which means that movement or motor imagery of different body parts is reflected as a decrease in SMR in the regions along the primary sensorimotor cortex. This important feature enables classification of different planning/imagining of different movements, which is served as the basis for SMR-based BCIs.

EEG Signal Processing

A BCI system, typically, consists of different components including the following two main building blocks: (i) A signal recording or brain imaging modality such as EEG, Near Infrared Spectroscopy (NIRS), Electrocorticogram (ECoG), or Magnetic Resonance Imaging (MRI) to monitor brain activities, and; (ii) A signal processing module, which is utilized to extract meaningful information from the recordings of the selected imaging modality. Having the affordability, portability, and high temporal resolution taken into account as desired features of a brain imaging modality, EEG is considered as the prime choice and is widely deployed in various practical BCI system. Processing of EEG signals generally includes two major tasks, i.e., "*Feature Generation,*" and "*Feature Translation*". While the former is mainly concerned with the pre-processing and filtering of the EEG signals to extract informative features capable of describing the intended underlying phenomena, the latter deals with utilization of the extracted features for classification and discrimination of intended tasks (e.g., actions, mental states, and/or emotions). Next, these two categories are reviewed to better position our work's contribution within the existing literature.

- **Feature Generation:** Extraction/generation of proper and discriminative features from raw signal is of great importance in EEG studies. The features not only map the high dimensional raw data into a lower dimension while preserving the useful information but also provide enough flexibility to discriminate different mental tasks. However, due to the weak potentials and interference from different physiological activities and also environmental interferences, it is hard to detect/extract the exact electrical response of the brain to the mental task and consequently, it is hard to extract proper features.

  Among different processing techniques on sensorimotor activities, the Common Spatial Patterns (CSP) [52] is known to be an effective tool for extracting features for the MI tasks. The CSP forms spatial filters that result in a more precise detection of the ERD and the ERS waveforms. Consequently, the CSP technique focuses more on the channels which demonstrate higher weights of the ERD and ERS waveforms. In other words, CSP aims at maximizing the variance of one MI task while minimizing the variance of the other task [53]. The superior power of the CSP approach in discriminating MI tasks has motivated the researchers to further extend the method to enhance its classification performance. The superiority of the CSP approach in discriminating MI tasks has motivated several recent research works [54–59] extending it to further enhance the classification performance.

  Generally speaking, extensions to the CSP technique can be classified into two major categories: (i) Filter bank-based solutions such as Filter Bank Common Spatial Patterns (FBCSP) [54], regularized common spatial patterns (RCSP) [33,55,56], and Separable Common Spectra-Spatial Patterns (SCSSP) [57]), which decompose the

EEG signals into several frequency bands, and; (ii) The second group such as [59], focus on deriving a single optimized frequency band for each subject to boost the CSP performance. Reference [60] employed a filter bank approach consisting of 3 spectral filters with fixed ranges, hence referred to as a Ternary approach. The work in [54] showed that the performance of the CSP method drastically improves while the EEG signals are separated into different frequency bands and each frequency band is analyzed separately. This initiated several contributions in this regard, among them the FBCSP [54] method is the most successful one which deploys a number of frequency bands with deterministic limits to extract frequency-specific features. The superior power of this technique is still a source of inspiration for newer research works including SCSSP [57] which tries to improve the FBCSP method in terms of runtime. On the other hand, some works proposed deployment of an optimization technique as in Bayesian Spectra-Spatial Filter Optimization (BSSFO) [59] method, aiming to optimize the limits of a single frequency band in contrary to the deployment of several deterministic bands utilized by filter-bank methods. In the other hand, processing of the EEG signals based on the Riemannian geometry has recently received a huge attention and is providing compromising results. The applicability of the Riemannian geometry is based on the fact that the EEG signals are recorded from the scalp, which forms a curvature where the Euclidean geometry is not applicable.

- **Feature Translation:** Feature translation is the process of translating the extracted features to the class labels. In case of dealing with classification of two classes of data, binary classifiers such as Support Vector Machines (SVM) or Linear Discriminant Analysis (LDA) are utilized. These methods are inherently designed for two-class problems and have shown superior performance in a huge number of applications. On the other hand, when it comes to multi-class problems, these techniques are not directly applicable and new classifiers for multi-class problems should be deployed; either by devising new classifiers or modifying the current binary classifiers. Among the classification techniques which are specifically developed for multi-class cases, the random forest, K-Nearest Neighbors (K-NN), K-means and multi-class SVM could be named. On the other hand, the OVO and One vs. All (OVA) are among the techniques which extend the binary classifiers and fit them for the multi-class problems. In many applications, the OVA and OVO techniques could be served as the baseline for comparison of different multi-class classification techniques.

Deep Learning Methods in EEG Processing

Deep Learning (DL) methods present the state-of-the-art performance in various machine learning applications such as computer vision and natural language processing. In 2012, the Convolutional Neural Network (CNN) framework won the ImageNet competition [61] and reignited the interest in neural networks, i.e., the main building block of DL methods. Since then, convolutional architectures play an important role in vision-related applications and various modifications of their architecture are studied [62–67] to improve their performance. Beside computer vision tasks, DL and CNNs have recently been employed in other domains such as speech recognition, text understanding, and also in the BCI systems. Applicability of CNNs in BCI is based on the fact that CNNs are well suited to identify events or activations in spatially distributed data and are also capable of exploiting meaningful features to translate and classify events of interest. Analogously, Motor Imagery (MI) BCIs also look for a change of activity in certain areas of brain to identify the subject's intention in moving a limb and then translate the detected event into proper commands. Having discussed the similarity of the two domains, utilization of DL methods in EEG-based BCIs has not yet been fully investigated. High dimensionality of EEG data, correlation between the recording channels and the presence of noise and artifacts in the recordings are the main challenges in developing DL-based BCIs [68]. Typically, to attain the goal of employing DL methods in BCIs, preprocessing of the signals is crucial to remove noise and artifacts and reduce the dimensionality of the signals, while having minimum information loss. References [68–73] represent recent research works in the field of DL-based BCI. The work in [73] employed Multi-layer Perceptron (MLP) and CNN architectures to independently analyze EEG signals and the final decision on the class of the signal is achieved by averaging the votes of the two parallel networks. More recently, the work in [68], which is an extension to [73], employed the CNN architecture along with temporal representations of the signals to classify the EEG segments. It is shown that temporal representation of the EEG signals in terms of signal envelopes carries more information than the log-energy features which are typically used in EEG classification. This work has achieved the state-of-the-art results on BCI competition IV_{2a} dataset.

2.1.2 Assistive Technologies

The increasing trend in the population of seniors has proportionally increased the number of individuals affected by age-related neurological disorders such as PD, ET, and their common motor symptoms such as PHT [6–8]. PHT is a common upper-limb motor symptom of several age-related neurological movement disorders and is described as involuntary and pseudo-rhythmic movements [10] affecting coordination, targeting, and speed of intended motions [11, 74, 75].

Unlike physiological tremor, which is identified with low amplitude vibrations occurring within the spectral range of 6 to 14 Hz [76] and affects the performance of individuals in high precision tasks such as robotic surgery [77], PHT represents higher amplitude motion occurring in the broader frequency range of $3-14$ Hz [78]. The repetitive and oscillating nature of PHT differentiates itself from other involuntary movement disorders such as chorea, athetosis, ballism, tics, and myoclonus [79]. Upper-limb tremor significantly limits individuals in performing the ADLs [80]. Thus, during the last decade, several techniques and technologies have been proposed in both rehabilitation and assistive domains [81–93] to compensate for the involuntary movement while promoting the voluntary component of motion. In fact, on-line and accurate estimation/extraction of PHT is of paramount importance to develop new assistive and rehabilitation technologies such as robotic rehabilitation systems and assistive robotic surgery. The extracted signals can be used in the design of technologies that aim to damp, counteract or compensate for the tremor. This may be done to assist or to rehabilitate motor control of patients using interactive robotic or wearable exoskeleton platforms. In other words, Accurate separation of the voluntary and the involuntary (PHT) components of the hand motion signal is of significant clinical and practical importance mainly due to the following main reasons:

(i) The tremor is considered as a critical measure for physicians to perform diagnosis and to recommend appropriate treatment therapies. Having a tremor extraction algorithm with high accuracy is, therefore, both beneficial and essential to differentiate common movement disorders such as Parkinson's disease and Essential Tremor, as in many cases they share similar characteristics.

(ii) In rehabilitation and assistive settings [94], accurate and real-time tremor extraction with minimum phase lag is crucially necessary to allow for proper delivery of assistive actions and guarantee the required level of safety.

Active compensation of tremor includes the following three steps: (a) Sensing and measuring the hand motion signal; (b) Estimation and extraction of the tremor out of the measurement signal, and; (c) Generation of control signals to counteract/compensate the tremor [95] while assisting the voluntary component of the motion. The estimation/extraction accuracy of Step (b) severely affects the performance of Step (c) as well as the overall performance of the systems depending on the accuracy of these estimates.

The accuracy of a tremor compensation technology (such as sophisticated wearable exosuits) rely significantly on the efficacy and spectrotemporal resolution of the algorithm, as inaccurate or slow extraction techniques do not allow for proper compensation. PHT consists of the following types of tremor: Rest Tremor (RT) occurring when a limb is relaxed and supported against gravity (commonly observed in PD [96]); Action Tremor (AT) occurring during voluntary contraction of muscles and classified into the subcategories of postural, kinetic, and isometric [97]. Postural and kinetic tremors are commonly observed in ET patients [96]. While postural tremor occurs when an individual voluntarily maintains a position against gravity, such as an outstretched arm, kinetic tremor occurs when a voluntary movement is performed. On the other hand, isometric action tremor exists during muscle contraction against a rigid stationary object such as grasping a solid object that blocks the motion of the limb and changes the length of muscles. It is worth mentioning that although PD tremor is typically characterized by unilateral rest tremor in the spectral range of $4-6$ Hz and ET patients commonly show symmetric postural and kinetic tremor in the range of $4-8$ Hz, there are many atypical cases (for example PD patients having action tremor and ET patients having rest tremor) that PD and ET share overlapping features [98].

PHT therapies such as oral medication, therapeutic lesions, Gamma-knife radiosurgery, and DBS [84] focus on alleviating tremor severity and improving arm functionality [32,99, 100]. Although efficacy is reported for these PHT interventions, about 25% of patients are unresponsive or experience short-term suboptimal response [80,101,102], and adverse side effects are also commonly observed. The severity and characteristics of PHT are assessed and monitored [8,103–113] through recording and processing of hand motions in clinical settings while performing different tasks. This information has also been used to tailor dosing and regimen of therapy, such as Botulinum Toxin type A (BoNT-A) injections [114]. However, a major remaining challenge in assessing action tremor is the processing and separation of voluntary and involuntary components, which is not accurate using conventional approaches. Thus, some of the recently-developed techniques and therapies (e.g. BoNT-A therapy) [114–118] that finely tailor dosing and muscle selection for targeted therapy based on accurate signal monitoring, would not be feasible for patients with prominent action tremor.

Recently, robotic rehabilitation and assistive technologies [119–123] have attracted a great deal of interest due to their promising performance in compensating the involuntary tremor in both rehabilitation and assistive settings. Such technologies [90, 119, 121, 124] are mainly developed to remove (damp out or compensate) the tremor and assist patients in performing their voluntary movements [91,95,120,125]. However, the performance and efficacy of such technologies is directly linked to the accuracy of tremor estimation, which is a nonstationary, nonlinear, and uncertain signal processing challenge. Thus, accurate PHT elimination (and possibly prediction) is important for both robotic rehabilitation and assistive technologies, and also for different clinical applications. This will allow proper delivery of the expected assistive actions at the right time of oscillation, and also guarantees the required level of safety. In particular, for these settings, it is essential to have zero or minimal phase lag in the estimation and extraction of tremor to effectively and simultaneously generate a counteracting force field. It is shown [85] that a phase-lag as low as 20 *ms* in the signal filtering part noticeably degrades the system performance. As the frequency content of pathological tremor can be close to that of the voluntary component, having zero or minimal phase-lag is a major challenge. However, this would be feasible if a robust and generalizable model of tremor is designed that can encapsulate nonlinear temporal dependencies between sub-movements during task performance.

Despite the crucial need for PHT elimination techniques and the numerous research advancements on this topic [126–130], there is an unmet need for a reliable, adaptable, and generalizable processing framework that can be directly translated to clinical settings for estimation, extraction, and prediction of action tremor with high spectrotemporal resolution. Among previously published studies, Band-limited Multiple Fourier Linear Combiner (BMFLC) [126] and Extended-BMFLC (EBMFLC) [81] frameworks demonstrate the most successful current state of the available approaches. The FLC-based methods, e.g., BMFLC and EBMFLC, are aimed at deriving linear mixing models for the spectral contents of the motion signal to distinguish and separate the two components. The BMFLC employs fixed predefined values to identify the spectral range of PHT and is aimed at attenuating those contents to derive the voluntary component. The EBMFLC, on the other hand, tries to adaptively identify and remove the spectral range of the PHT, which has shown superior performance compared to its counterpart. Existing methods including EBMFLC, share a similar characteristic by assuming that the spectral contents of voluntary and involuntary components are distinct and one could be derived by removing the other from the measurement signal. However, this assumption is not always realistic and has resulted in limited performance of the designed techniques and also hindered their clinical translation. In fact, the frequency contents of the voluntary and tremor components are not completely distinct from each other, and the overlapping of spectral contents is quite natural. Thus, considering all high-frequency components as tremulous motion results in inferior performance of the technique, which may fail to follow voluntary changes in the direction and frequency context of motion. Such behaviour can dramatically degrade the performance of any assistive technology that is supposed to react to the changes in motor intent, in an agile manner.

This can significantly affect the estimation of involuntary components (through misinterpretation as involuntary motion the high frequency content of voluntary motion caused by nonlinear and nonstationary changes). This can directly affect the regimen of advanced new therapies that are tailored based on such measures. We believe that the above-mentioned issue is an important contributing factor for the limited performance of previous methods.

2.1.3 Diagnosis of Parkinson's Disease vs. Essential Tremor

As discussed earlier, the population of seniors (aged 60 and above) is estimated to rise from 962 million in 2017 to 2.1 billion by 2050, and 3.1 billion by 2100 [2], which proportionally increases the population of the individuals affected by age-related neurological disorders [6–9]. To better manage the growing population of patients, specialized and advanced technologies are required to prevent, control, and cure age-related neurological diseases. PD and ET are among the common age-related neurological disorders, which respectively occur at the prevalence rate of $\sim 2\%$ and $\sim 4.5\%$ for individuals over 65 years old [98,131,132]. PD and ET share some common symptoms, including PHT, which affects coordination, targeting, and speed of voluntary motions [11] by the involuntary and pseudo-rhythmic movement of limbs [10]. There are various categorizations of PHT but two types are very common, namely "Rest Tremor" and "Action Tremor", where the latter is further classified into three subcategories of postural, kinetic, and isometric tremors [97, 133]. Rest tremor occurs when a limb is in a resting state and is supported against gravity, while action tremor occurs in case of voluntary contraction of muscles in a limb. Postural, kinetic, and isometric tremors are respectively observed when a patient maintains a position against gravity (such as stretched-out arms), performing a voluntary action, and contraction of muscles against a rigid object, respectively. While both PD and ET patients develop tremors, there are characteristic differences, potentially allowing differentiation of these two diseases. More specifically, PD is typically characterized by unilateral rest tremor in the spectral range of $4-6$ Hz [98, 134], whereas ET patients commonly show symmetric postural and kinetic tremor in the range of $4-8$ Hz [134, 135].

Although PD and ET could be characterized by their type of tremor, they also share overlapping features, especially in the early stages of the diseases [98, 136]. For instance, both rest and action tremors are observed in PD and ET patients to the extent that 46% of ET population show rest tremors [137] and up to 90% of PD patients have action tremor [133, 138,139]. In addition, a considerable number of ET patients show asymmetric hand tremors [133, 139, 140], given the fact that asymmetry of PHT is sometimes seen as a key signature of PD. In addition, the age range in which patients start to develop symptoms of PD or ET is not significantly different, further complicating the differential discrimination of the two diseases [141, 142]. The aforementioned overlapping features of PD and ET makes it significantly challenging to conduct differential diagnosis [136, 141, 143, 144], to the extent that 37% of ET patients are misdiagnosed and most of them are diagnosed as PD. Several studies [143, 144] have shown that 15% to 35% of patients with other movement disorders are also misdiagnosed as PD. Misdiagnosis of PD and ET can adversely affect the outcome of clinical trials and results in suboptimal treatment and faulty prognosis [136, 141]. This would be more concerning if misdiagnosed patients are prescribed to take dopaminergic medications for a long period of time. Finally, it is interesting to note that the correct PD diagnosis rates of movement disorder experts and neurologists are 80% and 74% [136], respectively. Consequently, it is of paramount importance to develop and devise advanced diagnosis techniques to significantly avoid such misdiagnosis of PD and ET.

In order to decrease the misdiagnosis rate, in the literature, some sophisticated technological solutions have been proposed to monitor symptoms of patients and track the correlated physiological phenomena. In this regard, recently, Positron Emission Tomography (PET) has been employed to study brain functions in the case of neuro-degenerative disorders, including PD. Scanning Dopamine Transporters (DAT) with PET [145,146] or Single Photon Emission Tomography (SPECT) have been recently considered as the gold standard (according to references [147,148]) for differential diagnosis of PD from ET, especially for ambiguous cases [149]. However, due to the expensive and time-consuming nature of PET and SPECT technologies and the need for injecting radioactive-labeled tracers, they are not widely employed [150] and thus investigating alternative diagnostic procedures is of high importance. In this regard, basic time-series analyses of tremorous motion of the limbs, and electrical activity of muscles are suggested as potential biomarkers that can help with the diagnosis [138,150]. The frequency contents of such recordings are known to reveal useful information for discrimination of PD and ET [97, 149]. Thus, various signal processing and Machine Learning (ML) techniques are investigated for such analysis of hand motion recordings of patients to better identify and discriminate the underlying characteristics, and assess the associated severity index.

To use time-series recordings for differentiating PD patients from those with similar symptoms but with different diagnosis, several classification schemes are developed over recent years in the literature [134, 151–153], including statistical signal processing [154], SVM [155], Naive Bayes classifiers, Nearest Centroid Classifier (NCC), Random Forest (RF) [156], Decision Tree (DT), and LDA [157, 158]. More recently, DL methods, which are considered as a subcategory of ML techniques and present methodologies to design multi-layer Artificial Neural Networks (ANN) are employed to analyze the tremor signals [159, 160]. The main benefit of DL methods compared to classical approaches is their independence from expert-defined features to grasp the underlying patterns of data. A meaningful representation of the signals is formed by a DL model when numerous training examples are being observed by the network to minimize a predefined cost function (e.g., classification error). Carefully crafted DL frameworks have shown superior performance in several practical applications and have ignited a great surge of interest in applying them to many different problems. However, the data-hungry nature of the DL techniques demands large datasets, which can represent a broad and clear image of the studied phenomenon and can help the network grasp a generic image of the characteristics of the two diseases from the tremor recordings. In fact, large datasets are required to grant an acceptable degree of the generalization to a neural network [161] to be securely deployed in real-world applications. Table 2.1 summarizes the research works on analysis of time-series recordings of tremorous limbs for diagnostic purposes, along with their achieved accuracy.

A growing surge of interest is observed in deploying DL methods, more specifically CNNs, in analyzing time-series recordings of tremorous limbs. In CNNs, a number of initially-randomized kernels (filters) are designed and convolved with raw data to capture the underlying patterns [162]. Commonly, several filter layers (hence the term deep learning) are stacked to derive a new informative representation. Technically, CNNs have outperformed computer-level and human-level performance in image [61] and speech recognition [163], justifying the growing trend of their application in other fields, e.g., tremor assessment [159, 160, 164–166]. The superior performance of CNN in the analysis of tremor recordings could be contributed to the fact that CNNs, as a subcategory of data-driven ML algorithms, do not require hand-crafted and expert-defined features to understand the studied phenomena and the inference is made by observing a considerable number of training examples and optimizing the parameters of neural network based on minimizing a predefined cost function. One of the main challenges of data-hungry deep neural networks is the interpretability of the results. Although high performance can be achieved, sometimes the network may focus on hidden biases in the dataset. For example, if the signals of the two conditions are recorded using two different machines (with particular spectrotemporal characteristics), a black box neural network may learn how to differentiate between the recording of the two machines, instead of the characteristics of the two conditions. To avoid that, researchers constantly evaluate all possible biases in the dataset, but without an interpretable solution this is always a concern. To address this issue and to encode a degree of transparency and interpretability in the machine learning models, a new set of techniques, referred to as explainable AI or XAI for short, are developed.

2.2 Deep Learning Methods

Machine learning is defined as a study of statistical and mathematical models, which enable a computer to capture the behaviour of a certain phenomenon without explicit instructions. Conventional machine learning methods are based on hand-crafted and user-engineered techniques developed to transform and represent raw data in a format which is perceivable by mostly-linear, or linear-in-parameter mathematical models. Performance of traditional machine learning methods, however, is normally restricted due to their limited modeling/learning capability. In addition, conventional machine learning methods require domain expertise and careful system design in order to have an acceptable performance. Therefore, representation learning methods [169] are introduced and developed such that the intrinsic patterns of input data are automatically inferred and extracted.

The successful application of data-driven machine learning techniques, e.g., deep learning methods, in image and speech processing domains has motivated their adoption in various other domains, including biomedical engineering. The superior learning capacity of neural networks, which is not based on user-engineered features and the data-driven methodology of optimizing the network parameters, depicts a great potential in employing these methods to develop ADs. In the thesis, Convolutional Neural Networks and Recurrent Neural Networks, which are respectively specialized in processing the spatially distributed data and sequences of data, are employed to process the EEG signals recorded from the human brain (multi-channel and spatially distributed) and the hand motion signals (time-series and sequential). In addition, deep metric learning frameworks are also developed to better suit the deep learning techniques for medical datasets with limited sizes. In what follows, each of these techniques are discussed.

Table 2.1: Literature review of the recent works in tremor signal analysis.

Reference	Goal	Dataset	Method	Results
Hossen *et. al.* [149] (2013)	ET/PD Classification	Accelerometer data, [19 PD, 21 ET] for training and [20 PD, 20 ET] for testing	Statistical Signal Characterization performed on the spectral domain of tremor signals	Accuracy = 90%
Ghassemi *et. al.* [134] (2016)	ET/PD Classification	Electromyogram and accelerometer data, [13 PD, 11 ET] for training and testing	Classification of Wavelet features with Support Vector Machines (SVM)	Accuracy = 83%
Eskofier *et. al.* [159] (2016)	Identify bradykinesia in PD patients	Accelerometer data, [10 PD, 960 assessments]	Convolutional Neural Network	Accuracy = 90.9%
Brzan *et. al.* [167] (2017)	ET/PD Classification	Electromyogram data [27 PD, 27 ET] for training and testing	A set of statistical and physiological features classified with decision tree	Accuracy = 94%
Di Biase *et. al.* [141] (2017)	ET/PD Classification	Accelerometer data, [16 PD, 20 ET] for training and [55] for testing	Analysis in spectral domain	Accuracy = 92%, Sensitivity = 95%, Specificity = 95%
Barrantes *et. al.* [168] (2017)	ET/PD/Healthy Classification	Accelerometer data, [17 PD, 16 ET, 12 healthy, 7 unknown]	Spectral analysis of the signals	Accuracy = 84.38%
Molparia *et. al.* [98] (2018)	ET/PD Classification	Accelerometer data and genetic profiles, [40 PD, 27 ET] for training and testing	Statistical properties of signal along with genomics data	Sensitivity = 76%, Specificity = 65%
Kim *et. al.* [165] (2018)	PD severity scoring	Accelerometer and gyroscope data, [96 PD]	Convolutional Neural Network	Accuracy = 85%
Camps *et. al.* [166] (2018)	Detecting the freezing of gait in PD	Accelerometer, gyroscope, and magnetometer, [21 PD]	Convolutional Neural Network	Accuracy = 90%
Zheng *et. al.* [160] (2019)	ET severity scoring	Accelerometer data, [20 ET]	Convolutional Neural Network	Accuracy = 85.44%

2.2.1 Convolutional Neural Networks

Convolutional architectures [61, 170] play an important role in vision-related applications and various modifications of their architecture are studied in [62–67] to improve their performance. Besides the computer vision tasks, DL and CNNs have recently been employed in other domains such as speech recognition, text understanding, and also in the biomedical signal and image processing problems [171]. The main advantage of CNNs over simple neural networks is the shared parameter strategy for the convolutional kernels across all of the data points in the input data, which drastically decreases the number of trainable parameters and enhances the robustness of the network to variations in the input data. The properties of the kernels are learned through the training process and each kernel becomes specialized in detecting one feature in the input data. In fact, CNNs are stack of convolutional, pooling, and fully connected layers, where the convolutional layers are responsible for learning and extracting features by applying trainable filters on the input. In general, in a CNN with L layers, the convolutional operation of the layer l, for $(2 \leq l \leq L)$, on the output from the layer $l-1$, denoted by $\boldsymbol{Y}^{(l-1)}$, results in the pre-activation output $\boldsymbol{X}^{(l)} = [X_{i,j}^{(l)}]$ given by

$$X_{i,j}^{(l)} = \sum_{a=0}^{M-1}\sum_{b=0}^{M-1} W_{a,b}^{(l)} Y_{i+a,j+b}^{(l-1)}, \tag{2.1}$$

where M is the size of kernels, and $\boldsymbol{W}^{(l)} = [W_{a,b}^{(l)}]$ is the kernel matrix containing the CNN weights to be learned during the back propagation. Term $\boldsymbol{X}^{(l)}$ will, accordingly, go through an activation function, denoted by $\sigma(\cdot)$, resulting in the final output $\boldsymbol{Y}^{(l)}$, as follows

$$Y_{i,j}^{(l)} = \sigma(X_{i,j}^{(l)}). \tag{2.2}$$

Pooling layers act as sub-sampling parts to make the network more translational invariant. Finally, the fully connected layers have the same functionalities as a simple neural network. For the case of tremor signal analysis, we feed the convolutional filters with the spectrogram of the signals, as they reveal a $2-$dimensional representation of the temporal and spectral features and are perfectly suited to be analyzed by $2-$dimensional kernels.

2.2.2 Recurrent Neural Networks

Recurrent Neural Network (RNN) models are a subcategory of representation learning methods [172] which are specialized in analyzing sequential data and detecting long-term and short-term temporal dependencies in signals based on nonlinear embedded memory. In RNN models, at each time instance, a combination of input sequence and hidden state vector of the previous time instance are analyzed together to update the state vector and pass it to the next time instance. This process continues until the whole sequence of data is analyzed and a meaningful representation is formed. RNNs have various designs to fit different applications, e.g., sequence to sequence RNNs are employed for machine translation tasks, and sequence to single-output RNNs are employed for classification tasks. Since in the thesis, RNNs are employed in the design of AD to process the kinematics of hand motion and the goal is to translate a tremor-contaminated sequence to a sequence representing the voluntary component of hand motion, we employ the sequence-to-sequence architecture. A recurrent network with basic hidden cells is formulated as

$$\boldsymbol{h}(t) = \mathrm{f}\big(\boldsymbol{b} + \boldsymbol{W}\boldsymbol{h}(t-1) + \boldsymbol{U}\boldsymbol{m}(t_1:t)\big), \tag{2.3}$$

$$\text{and} \qquad \hat{\boldsymbol{y}}(t) = \mathrm{softmax}\big(\boldsymbol{c} + \boldsymbol{V}\boldsymbol{h}(t)\big), \tag{2.4}$$

where $\boldsymbol{m}(t_1:t) = [m(t_1), \ldots, m(t)]^T$ is the hand motion signal from time $(t_1 < t)$ to time t as the input sequence of the network; $\boldsymbol{h}(t)$ is the hidden feature vector; $\boldsymbol{b}$ is the bias vector for the input nodes; $\boldsymbol{W}$ is the weight matrix for hidden-to-hidden connections; $\boldsymbol{U}$ denotes the input-to-hidden weights of the RNN; $\boldsymbol{c}$ is the bias vector for the output nodes; $\boldsymbol{V}$ denotes the weight matrix for hidden-to-output connections; and $f(\cdot)$ denotes a nonlinear function. We note that the weights and biases in Eqs. (2.3)-(2.4) are optimized during the training phase.

Using an RNN it can be expected that the output of the network for the very initial input samples is inaccurate and as the information propagates across the network and more samples of the input sequence are analyzed, the output becomes more accurate. Consequently, the output sequence becomes more reliable (in terms of its similarity to the ground truth signal) after a transient phase of initial inputs. Since the objective of the thesis is to develop an online and offline tremor extraction framework, we structure the processing pipeline in a bidirectional format which employs two parallel sets of recurrent cells for the two processing schemes. In other words, forward cells are employed for online (predictive) processing of the input sequence where we need maximum accuracy of estimation for the last samples of the output sequence. Backward cells, on the other hand, are employed for offline processing of measurement signals, following the same logic for the forward cells. It is worth mentioning that in the utilized architecture, the base of which has been named in the literature as Bidirectional RNN (BRNN) [173], the forward and backward hidden cells are usually followed by a mixing matrix which merges the outputs of the two paths. However, in this work, through an architectural modification of the model, the BRNN kernel is applied without the mixing matrix. This is done since the ultimate goal is to have two separate processing pipelines for both online and offline applications.

A common problem with the basic RNN model described in Eqs. (2.3)-(2.4) is its weakness in capturing long-term patterns of the input sequence. This shortcoming is pronounced when long sequences of data need to be processed or when the input sequence encapsulates nonstationary patterns, which is the case for PHT. Moreover, training of these networks is very critical since the problems of vanishing or exploding gradients are prevalent. To address these issues, the Gated Recurrent Unit (GRU) cells [174] based on "reset" and "update" gates were developed, where the reset gate determines the degree of dismissing old information and considering the data from input in the current time, and the update gate defines the degree of updating a hidden state based on the newly arrived data [175]. Therefore, we can update Eq. (2.3) as follows

$$\begin{aligned}
\boldsymbol{r} &= \sigma\big(\boldsymbol{U}_r\boldsymbol{m}(t_1\!:\!t)+\boldsymbol{W}_r\boldsymbol{h}(t-1)\big), && (2.5)\\
z &= \sigma\big(\boldsymbol{U}_z\boldsymbol{m}(t_1\!:\!t)+\boldsymbol{W}_z\boldsymbol{h}(t-1)\big), && (2.6)\\
\tilde{\boldsymbol{h}}(t) &= \mathrm{ReLU}\big(\boldsymbol{U}\boldsymbol{m}(t_1\!:\!t)+\boldsymbol{W}(\boldsymbol{r}\odot\boldsymbol{h}(t-1))\big), && (2.7)\\
\text{and}\qquad \boldsymbol{h}(t) &= (1-z)\boldsymbol{h}(t-1)+z\tilde{\boldsymbol{h}}(t), && (2.8)
\end{aligned}$$

where the reset gate is denoted by r and the update gate is denoted by z. Consequently, their corresponding weights are denoted by $\boldsymbol{U}_r, \boldsymbol{W}_r$ and $\boldsymbol{U}_z, \boldsymbol{W}_z$, respectively. The term σ denotes a logistic sigmoid function. As PHT can be a highly dynamic and nonstationary phenomenon, GRU cells are utilized to better capture the long-term behavioural variations of the hand motion signal. One can construct a deep neural architecture by stacking several layers of RNN such that the output of one layer is provided as the input to the next layer, to enhance the learning capacity of the network and more accurately discover the underlying patterns of the input sequence.

2.2.3 Deep Metric Learning

Metric learning [176–178] refers to a category of machine learning techniques to optimize an encoding network. The ultimate goal is to maximize the inter-class similarity/distance, with respect to certain objective functions, while minimizing the intra-class distance between extracted embeddings. The profound capacity of such methods in handling problems with dynamic number of classes and their generalization capabilities to model underlying distribution of data with limited training samples have ignited a growing surge of interest in their adoption into various applications including but not limited to face recognition [179], signature verification [180], and few-shot learning [181, 182]. Besides, the promising performance of DL methods [61, 183] in estimating the distribution of data and capturing the underlying patterns has enabled extraction of high-level semantic embeddings from data. Deep Metric Learning (DML) [184] leverages the capacity of DL methods in extracting distance preserving embeddings and has surpassed the performance of conventional methods in different applications including face verification [185, 186], zero-shot classification [187, 188] and image retrieval [189–192].

In the context of deep metric learning, at one hand, the ongoing focus is on investigating various network architectures [193, 194] and training schemes [192, 195, 196] to further boost their performance. On the other hand, devising novel objective functions to measure the similarity/distance of embeddings to ensure a fast convergence and optimal global minimum search [182] is of paramount importance. The latter has received a growing attention in recent years and is the focus of this paper. In this regard, the "*Contrastive Loss*" function [197] was primarily introduced, which measures the Euclidean distance between one pair of embeddings to decide if they represent the same class or not. Later on, a better performance in terms of generalization and convergence speed was achieved by "*Triplet Loss*" [176, 187], which utilizes a query sample, a positive instance (same-class), and a negative instance (different-class) to locally grasp the orientation of positive and negative instances with respect to each other. In the triplet loss, an encoding neural network is optimized such that the distance between positive and negative samples is simultaneously minimized and maximized, respectively. However, since the two aforementioned methods employ only one negative instance in each update of the network, both methods suffer from slow convergence and poor local optima [196]. This shortcoming was later on addressed by "*Multi-class N-pair loss (MCNP)*" [196], which for a N-class problem incorporates $N-1$ negative instances from different classes to train the encoder network. Despite all these progress and although the MCNP outperformed its counterparts, the following key issues remained to be addressed:

- First, ignoring the distance between negative instances remains a shortcoming of the MCNP limiting its convergence, as the optimization of distance for negative instances is not assured.

- In addition, the aforementioned methods including the MCNP are based on the Euclidean distance and dot product of the embeddings, which are unbounded metrics increasing the risk of large variances in the network [198]. This issue could also lead to the sensitivity of the model to covariate shift, which therefore results in poor generalization and slow convergence of the model [199, 200].

- While normalization methods can be employed to constrain the variance and control the convergence, however, normalization can result in a non-convex loss formulation [201].

Capitalizing on the above mentioned issues, Cosine similarity could be used as an alternative metric, which is only influenced by the alignment of embeddings rather than their magnitude. Cosine similarity is commonly used as a metric in information retrieval and data mining to measure the similarity of two documents [202] for classification or clustering of high dimensional feature spaces. Some works have proposed employing cosine similarity coupled with a variant of softmax loss in different scenarios [179, 203–208] and promising performances are achieved. However, this combination typically imposes a number of hyperparameters in the loss function resulting in the instability of training process [209]. In other words, performance of such techniques relies on careful parameterizations of the hyparparameters, which requires a significantly large number of experimentation.

Let $\boldsymbol{X} \in \mathbb{R}^{L_1 \times \ldots \times L_M}$ be a M dimensional tensor, the input to a neural network and $\boldsymbol{o}^X$ be its extracted embedding of length K. Consider a N-class problem with input $\boldsymbol{X}$ belonging to Class C_i, for $(1 \leq i \leq N)$. An instance from the same class as that of $\boldsymbol{X}$ is called a positive instance $\boldsymbol{X}^{\mathscr{P}}$, i.e., $\boldsymbol{X}^{\mathscr{P}} \in C_i$. Instances from other classes, C_j, for $(1 \leq j \leq N$ and $j \neq i)$, are referred to as negative instances denoted by $\boldsymbol{X}^{\mathscr{N}}$.

In contrastive learning [197, 210], the distance between the embeddings of $\boldsymbol{X}$ and any other instance is calculated as

$$l^m_{cont} = 1\{\text{if same class}\}||\boldsymbol{o}^X - \boldsymbol{o}^{\mathscr{P}}||^2_2 + 1\{\text{if different class}\}\max(0, m - ||\boldsymbol{o}^X - \boldsymbol{o}^{\mathscr{N}}||_2)^2, \tag{2.9}$$

where m denotes the minimum distance margin between the positive and negative instances. The triplet loss [176, 187, 211], on the other hand, tries to enhance the generalization of the model by simultaneously comparing the query example with one positive and one negative instance, as in

$$l^m_{tri} = \max(0, ||\boldsymbol{o}^X - \boldsymbol{o}^{\mathscr{P}}||^2_2 - ||\boldsymbol{o}^X - \boldsymbol{o}^{\mathscr{N}}||^2_2 + m). \tag{2.10}$$

Similar to the contrastive learning, the hyperparameter m in triplet loss imposes a margin between the distances of positive and negative instances. Although the two aforementioned loss objectives are well-regarded in the literature, they both suffer from slow convergence and require expensive data mining to provide proper pairs or triplets for a faster training [187, 211]. To address these issues, Multi-class N-pair loss [196] was proposed, which simultaneously compares the distance between query example and positive instance with multiple negative instances from all of the other classes. This objective function is formulated as follows

$$l_{mcnp} = \log\Big(1 + \sum_{j=1}^{N-1} \exp\big({\boldsymbol{o}^X}^T \boldsymbol{o}^{\mathscr{N}_j} - {\boldsymbol{o}^X}^T \boldsymbol{o}^{\mathscr{P}}\big)\Big), \tag{2.11}$$

where superscript T denotes transpose operator. The objective function in Eq. (2.11) provides a better generalization and faster convergence by incorporating negative instances from all of the other classes and also relaxes the dependency to hyperparameter m. However, it employs dot product of embeddings, which increases the risk of large variances in the network and deteriorates the sensitivity of the model to covariate shift. This completes a brief review of widely-used objective functions in deep metric learning.

2.3 Wavelet Transformations

It is known that most of the local, transient and intermittent components of the signals get disappeared in frequency analysis methods such as Fourier Analysis due to averaging behavior of the methods. The spectra-temporal signal analysis methods, on the other hand, provide a more clear representation of the characteristics of signals in both temporal and spectral domains [212]. A number of spectra-temporal signal analysis methods have been investigated in literature including the Short Time Fourier Transform (STFT), Wigner-Ville Transform (WVT), Choi-Williams Distribution (CWD) and the Continuous Wavelet Transform (CWT). Among these methods, CWT has shown to be more favorable for researchers due to its less requirements for analysis and its allowance for high resolution for high frequency signals.

A wavelet transform is representing a signal into different scales and dilations of a "finite-length" and "fast-decaying" oscillating waveform known as a wavelet function (or mother wavelet) and scaling function (or father wavelet). Mother wavelet is in fact a band-pass filter which extracts the frequency contents of signal in its specific frequency band. On the other hand, the father wavelet is a low-pass filter and preserves the slow dynamics of the signals.

Mathematically speaking, a mother wavelet $\psi(t)$ should be a square integrable function, which satisfies the *admissibility condition* [213] which is given by

$$0 < c_\psi = \int_{-\infty}^{\infty} \frac{|\hat{\psi}(\omega)|}{|\omega|} < \infty \tag{2.12}$$

where $\hat{\psi}(\omega)$ is the Fourier transform of $\psi(t)$. The mother wavelet should preferably satisfy the regularity condition which requires that $\psi(t)$ be fast decaying or be nonzero only on a finite interval [213]. To obtain the wavelet transform of a signal, a group of different dilations and scales of the mother wavelet function, $\psi_{s,\tau}(t)$, are applied to the signal, which τ represents the dilation and s represents the scale of the mother wavelet function. $\psi_{s,\tau}(t)$ is derived as follows

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\psi\big(\frac{t-\tau}{s}\big), \quad s,\tau \in \mathbb{R}, s \neq 0. \tag{2.13}$$

The continuous wavelet transform of a signal, $\boldsymbol{x}(t)$, is calculated by,

$$\boldsymbol{T}_x(s,\tau) = \int_{-\infty}^{\infty} \boldsymbol{x}(t)\psi^*_{s,\tau}(t)dt. \tag{2.14}$$

Please note that $(.)^*$ denotes the complex conjugate. The *admissibility condition* for $\psi(t)$ ensures that the signal $\boldsymbol{x}(t)$ could be completely reconstructed from $\psi_{s,\tau}(t)$, according to the following formula

$$\boldsymbol{x}(t) = \frac{1}{c_\psi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\boldsymbol{T}_x(s,\tau)\psi_{s,\tau}(t)\frac{d\tau ds}{s^2}. \tag{2.15}$$

Since the values of s and τ vary over a continuous range of numbers in $\mathbb{R}$ domain, we observe redundancy in calculation of the wavelet coefficients. So in practical applications the values of s and τ are able to vary over a discrete range of numbers in the scale-time plane. This discretization leads to formation of *discrete wavelet* which instead of s and τ deploys discrete values of j and k for scaling and dilations, respectively. $\psi_{j,k}$ is defined as

$$\psi_{j,k}(t) = s_0^{-j/2}\psi(s_0^{-j}t - k\tau_0), \quad j,k \in \mathbb{Z}, \tag{2.16}$$

where $s_0 > 1$ and τ_0 are fixed dilation and translation factors [213]. The so called term of *dyadic sampling* is related to when the parameters are set to $s_0 = 2$ and $\tau_0 = 1$. The Discrete Wavelet Transform (DWT) is then defined by

$$\boldsymbol{T}_x(j,k) = \int_{-\infty}^{\infty}\boldsymbol{x}(t)\psi_{j,k}^*(t)dt. \tag{2.17}$$

If the family of wavelets $\psi_{j,k}(t)$ form an orthogonal basis, the signal $\boldsymbol{x}(t)$ could be recovered from its *discrete wavelet decomposition*, $\boldsymbol{T}_x(j,k)$, according to the following equation

$$\boldsymbol{x}(t) = \frac{1}{c_\psi}\sum_{j,k\in Z}\boldsymbol{T}_x(j,k)\psi_{j,k}(t). \tag{2.18}$$

It is worth mentioning that the DWT works best with the discrete signals which have fixed sampling rate.

As it is noted previously, wavelet transformation is representing a signal into different scaling and dilations of a group of mother and father wavelets. The father (scaling) function, which has also the same shape of the mother wavelet, represents an smoothed image of the signal and is defined by

$$\phi_{j,k}(t) = s_0^{-j/2}\phi(s_0^{-j}t - k\tau_0), \quad j,k \in \mathbb{Z}. \tag{2.19}$$

The $\phi_{j,k}(t)$ have the property that $\int_{-\infty}^{\infty}\phi_{0,0}(t) = 1$, where $\phi_{0,0} = \phi$.

As we discussed earlier, the mother wavelet acts as a high-pass filter, so its coefficients ($\boldsymbol{T}_x(j,k)$) represent the *details* of the signal, while the father wavelet could be modeled as a low-pass filter which provides the *approximations* of the signal. To derive the approximation coefficients of a signal, $\boldsymbol{S}_x(j,k)$, the father wavelet function should be convolved with the signal, as follows:

$$\boldsymbol{S}_x(j,k) = \int_{-\infty}^{\infty}\boldsymbol{x}(t)\phi_{j,k}dt. \tag{2.20}$$

The approximation coefficients at a specific scale j represent the discrete approximation of the signal at that scale. A continuous approximation of the signal at scale j is derived by summing a sequence of father wavelets at this scale factored by the approximation coefficients as follows

$$\boldsymbol{x}_j(t) = \sum_{k=-\infty}^{\infty} \boldsymbol{S}_x(j,k)\phi_{j,k}(t), \tag{2.21}$$

where $x_j(t)$ is a smooth, scaling-function-dependent version of the signal $\boldsymbol{x}(t)$ at scale index j. This continuous approximation approaches $\boldsymbol{x}(t)$ at small scales, i.e. as $j \to \infty$ [212]. A signal $\boldsymbol{x}(t)$ can then be represented using a combined series expansion using both the approximation coefficients and the wavelet (detail) coefficients as follows

$$\boldsymbol{x}(t) = \sum_{k=-\infty}^{\infty} \boldsymbol{S}_x(j_0,k)\phi_{j_0,k}(t) + \sum_{j=-\infty}^{j_0} \sum_{k=-\infty}^{\infty} \boldsymbol{T}_x(j,k)\psi_{j,k}(t). \tag{2.22}$$

2.4 Kalman Filtering

In this section we provide a brief introduction to the Kalman Filtering [214, 215] which is basically designed to estimate the state $X \in \mathbb{R}^n$ of a discrete-time process. The formulations provided here are chosen to be consistent with the original paper [214] for ease of understanding. The system in which Kalman filtering is studied operates based on the following linear stochastic difference equation:

$$X_k = \boldsymbol{F}X_{k-1} + \boldsymbol{B}U_{k-1} + \boldsymbol{w}_{k-1} \tag{2.23}$$

and the measurement $Z \in \mathbb{R}^m$ is defined as:

$$\boldsymbol{Z}_k = \boldsymbol{H}X_k + \boldsymbol{v}_k. \tag{2.24}$$

The terms $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are random variables which represent the process and measurement noise, respectively and are assumed to be independent of each other, obeying the following distributions:

$$p(\boldsymbol{w}) \sim N(0,Q), \tag{2.25}$$

$$p(\boldsymbol{v}) \sim N(0,R). \tag{2.26}$$

According to the system which is going to be modeled by Kalman equations, the *"process noise covariance* $\boldsymbol{Q}$*"* and *"measurement noise covariance* $\boldsymbol{R}$*"* might change over time, but in this work we assume them to be fixed over time. In Eq. (2.23) the matrix $\boldsymbol{F}$ which is $n \times n$ relates the state of the system in the previous time sample $(k-1)$ to the next time sample (k) independent from the input to the system (U) and the process noise (w). The values of matrix $\boldsymbol{F}$ could also vary over time but in this work we assume it to be constant. The matrix $\boldsymbol{B}$ which is $n \times l$ also relates the input, $(U \in \mathbb{R}^l)$, in step k to the states in step k. The matrix $\boldsymbol{H}$ is also $m \times n$ which relates the states to the measurement vector Z_k. The matrix $\boldsymbol{H}$ could also vary over time but here we take it as constant.

Kalman filters perform the estimation for the next time sample based on feedback control; which is estimating the next time sample and then receiving the noisy measurement for that time and then trying to reduce the error of estimation by adjusting the some parameters. Hence, the equations governing Kalman filters fall into two categories; ***time update*** equations and ***measurement update*** equations. The time update equations are responsible for *a priori* estimating the states in the next time sample based on the previous state and covariance matrix of process noise. On the other hand, the measurement update equations are responsible for incorporating the new noisy measurements into the *a priori* estimation process to enhance the accuracy of estimation. The measurement equations serve as the feedback loop for the system.

The time update equations can also be thought of as *predictor* equations, while the measurement update equations can be thought of as *corrector* equations; Hence, we can call the whole Kalman algorithm a *predictor-corrector* algorithm for solving numerical problems. The time update equations are

$$\hat{X}_k^- = \boldsymbol{F}\hat{X}_{k-1} + \boldsymbol{B}U_{k-1} \tag{2.27}$$

$$\text{and} \quad \boldsymbol{P}_k^- = \boldsymbol{F}\boldsymbol{P}_{k-1}\boldsymbol{F}^T + \boldsymbol{Q}. \tag{2.28}$$

As it is understood from the Eqs. (2.27) and (2.28), in time update stage the information of past state and process noise covariance are deployed to estimate the state in next time sample. The first task during the measurement update is to compute the Kalman gain, $\boldsymbol{K}_k$, which is provided in Eq.(2.29).

$$\boldsymbol{K}_k = \boldsymbol{P}_k^- \boldsymbol{H}^T (\boldsymbol{H}\boldsymbol{P}_k^- \boldsymbol{H}^T + \boldsymbol{R})^{-1}. \tag{2.29}$$

The next step is to actually measure the process to obtain Z_k, and then to generate an *a posteriori* state estimate by incorporating the measurement as in

$$\hat{X}_k = \hat{X}_k^- + \boldsymbol{K}_k \big(Z_k - \boldsymbol{H}\hat{X}_k^-\big). \tag{2.30}$$

The final step is to obtain an *a posteriori* error covariance estimate via

$$\boldsymbol{P}_k = \big(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}\big)\boldsymbol{P}_k^-, \tag{2.31}$$

when a pair of time and measurement update equations execute, the algorithm moves one sample ahead and uses the *a posteriori* estimations of the last step to provide the *a priori* estimation for the next time sample. The fact that kalman filter does not incorporate more than two time samples of the data is the main reason why the Kalman filtering is well regarded among other estimation techniques. In fact, the Kalman filter recursively takes the information from past measurements into account for estimating the next prediction.

2.5 Summary

In this chapter, a thorough literature review of ADs and BCI systems, as well as computational techniques for differential diagnosis of PD from ET was presented. In addition, an integrated use case of the novelties and contributions of the thesis in a technique referred to as “functional electrical stimulation for tremor suppression” was discussed. In the rest of the chapter, the required technical tools for the proposed methodologies in the thesis were presented. Next chapter will focus on the development of data-driven methods for EEG-based BCI systems, and will elaborate on the contributions of the thesis in that regard.

Chapter 3

Data-driven Methods for EEG-based Brain Computer Interfaces

3.1 Introduction

In Chapter 2, the urge for developing BCI systems that are capable of monitoring the human brain's activity and providing a new medium for individuals to communicate with other people or devices was discussed. Moreover, the undeniable role of a reliable BCI system in providing rehabilitation services to compensate for the lost functionalities of the patients affected by age-related neurological disorders was presented. It is also worth mentioning that a major bottleneck of the research works devoted to discovering and classifying the underlying and specific features of brain signals for different motor tasks is their limited performance in multiclass classification scenarios. This chapter introduces a statistical signal processing framework based on Bayesian optimization methods to derive subject-specific spectral and spatial filters for EEG processing. In addition, a novel classification scheme for multiclass problems is presented, which enhances the classification accuracy of BCI system over multiple MI tasks. Moreover, a DL-based EEG processing and feature extraction paradigm based on Siamese neural networks is presented, which can be conveniently merged and scaled up for multi-class problems and offers promising results for the MI classification problem. The chapter is concluded by introducing a dimensionality reduction framework based on Riemannian geometry and wavelet transformation, which can be effectively employed for EEG processing.

3.2 Bayesian Optimization Methods for EEG Processing

As discussed in Chapter 2, the filter bank methods have demonstrated promising results over different testing scenarios, however, a significant portion of the extracted features are not utilized in the classification step as some of the spectral ranges do not provide informative input for the classification, specially when fixed and general frequency bands are employed. On the other hand and to the best of my knowledge, the methods utilizing the optimized spectral filters, have only inspected the optimization of a single frequency band.

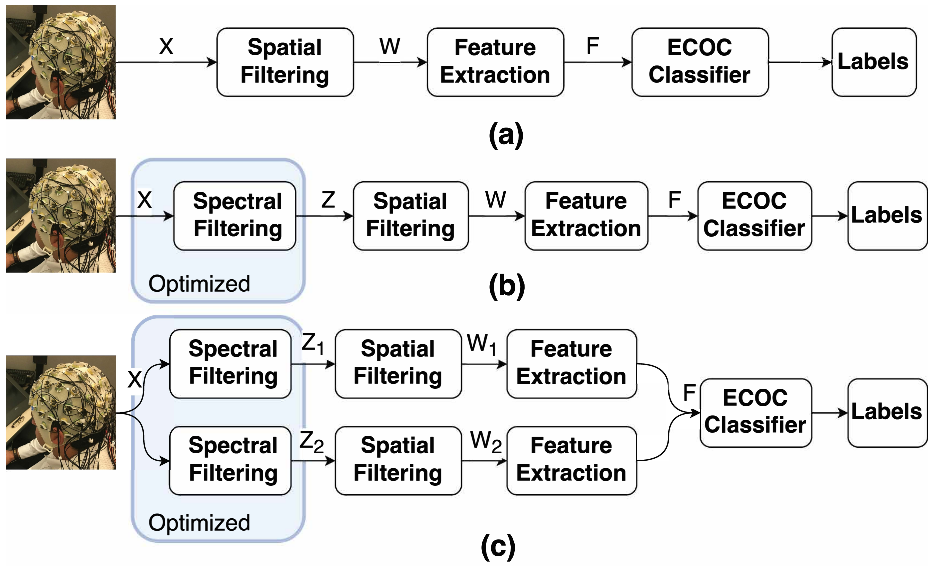


Figure 3.1: The ECCSP framework: (a) The ECCSP (b) The ECCSP-TB (c) The ECCSP-TB2B

We bridge the two worlds by devising an optimization framework for filter bank optimization. In addition, for multiclass classification problems, the efficacy of the classifier plays an important role in the overall performance of the processing pipeline and to this end, we introduce a new classification paradigm based on ternary numerical system shown to boost the overall performance.

To address the former foregoing drawback, in this section, a combination of filter banks and optimized techniques is proposed, which simultaneously takes advantage of the two solutions. To this aim, a Bayesian framework to optimize the cutoff frequencies of two bandpass filters is introduced. To address the latter drawback, the ECOC classifiers, which splits the classification problem into a number of binary ones is utilized. The ECOC [216] has been successfully used in other application domains, in particular, it has been effectively applied for text classification problems [217]. It is worth mentioning that despite the successful application of ECOC in other fields, it has not yet been used for EEG classification in the BCI systems, which is one scope of this chapter. In addition, this section introduces a modified version of the ECOC classifiers, which is referred to as Ternary-ECOC, to enhance the classification accuracy of BCI systems in multiple class scenarios. The overall theme of the proposed material in this section is demonstrated in Figure 3.1 and is briefly outlined below:

- **The ECCSP**, which is a combination of ECOC with CSP, where spatial filtering is performed via CSP technique and utilization of ECOC as the classifier is investigated (Figure 3.1(a)).

- **The ECCSP-TB**, which improves each of the two underlying components of the ECCSP as follows: (i) Modifies the ECOC algorithm and use its Ternary version, and; (ii) Incorporates a Bayesian framework to optimize the spatio-spectral filters prior to extracting the CSP features. The ECCSP-TB includes an optimization framework to obtain subject-specific spectral filters to enhance the classification accuracy,

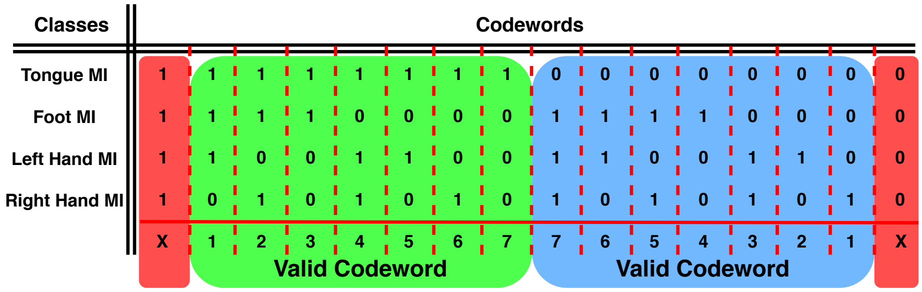


Figure 3.2: All possible 4-bit binary codewords for a 4 class problem. The columns denoted by "X" are not valid. Either of the two groups denoted by numbers from 1-7 could be employed as they are complement of each others.

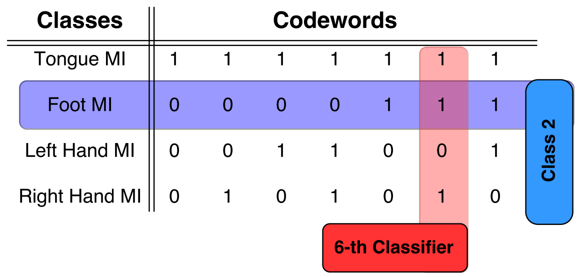


Figure 3.3: Generated ternary codewords for classifying 4 classes of data via TECOC approach.

and updates the ECOC classifier to the proposed ternary version to further improve the classification accuracy (Figure 3.1(b)).

- **The ECCSP-TB2B**, further improves the ECCSP framework by using two optimized frequency bands for the spectral filtering instead of the one used in the ECCSP-TB version. The ECCSP-TB2B version improves the spectral filtering stage of the ECCSP-TB framework to extract subject and frequency specific features leading to higher accuracy (Figure 3.1(c)).

In what follows, the ECCSP framework and its variants, ECCSP-TB and ECCSP-TB2B are discussed.

3.2.1 The ECCSP Framework

In this section, the main building block of the proposed ECCSP framework is presented, which is developed by coupling the ECOC classification approach with the CSP feature extraction methodology. In this work, my focus is on supervised learning (for classifying MI tasks) from N_{Trial} number of available training EEG epochs (trials) denoted by $\boldsymbol{X}_i \in \mathbb{R}^{N_{\text{ch}} \times N_{\text{t}}}$, for $(1 \leq i \leq N_{\text{Trial}})$. Trial $\boldsymbol{X}_i$ consists of N_{ch} number of EEG channel recordings, where N_{t} number of time samples are available per each channel. In other words, the processing is performed based on a training dataset denoted by $\{(\boldsymbol{X}_i, \Omega_i)\}$, where Ω_i is the correct label of the i^{th} epoch. A pre-processing step, including bandpass filtering (range of $0.5 - 100$Hz used by the BCI competition [218]) and notch filtering, is typically applied.

ECOC Classifiers

In ECOC classifiers, the multiclass classification problem is decomposed into a number of binary classification problems and the whole process is modeled as communication of a binary codeword over a noisy channel. In terms of a communication system, the received codeword is noise contaminated and possibly different from the originally transmitted one. In the classification problem, each bit of the codeword is derived by solving each of the binary classifiers and the noise in the communication channel is actually the misclassification error of each classifier. If proper binary codewords, in terms of their distance from each other, are assigned to different classes, then it is possible to handle a few bits of error and reconstruct the original codeword from the noisy one. In fact, if the codewords are separated with the Hamming distance of d, then it is possible to handle $\frac{d-1}{2}$ bit errors. For a multiclass problem with N_c classes, a binary coding matrix $\mathscr{C}$ with N_c rows each with N_Bits bits length needs to be created. Each row of the coding matrix $\mathscr{C}$ ($\mathscr{C}_n; 1 \geq n \geq N_\mathrm{c}$) represents the unique codeword associated with class n. Matrix $\mathscr{C}$ defines the number of binary classifiers to be trained ($\gamma^{(j)}; 1 \leq j \leq N_\mathrm{Bits}$) and each column of $\mathscr{C}$ represents the super-set shaping strategy for the each binary classifier. In this setting, the binary classifier $\gamma^{(j)}$ associated with the j^th column of the coding matrix $\mathscr{C}$ aims at distinguishing the trials which are labeled with 0 from the trials labeled with 1. This results in two super-sets for each classifier $\gamma^{(j)}$ denoted by $\boldsymbol{S}^{(j,0)}$ and $\boldsymbol{S}^{(j,1)}$, with their corresponding features which are extracted to be fed into the classifier. Training an ECOC, therefore, comprises of learning a set $\boldsymbol{\Lambda} = \{\gamma^{(1)}, \dots, \gamma^{(N_\mathrm{Bits})}\}$ of independent binary classifiers.

To properly form the codewords with enough separability in terms of Hamming distance, for $3 \geq N_\mathrm{c} \geq 7$, $N_\mathrm{Bits} = 2^{N_\mathrm{c}-1} - 1$ classifiers are required [219]. For $8 \geq N_\mathrm{c} \geq 11$, selection of a proper subset of columns from the exhaustive code by an optimization technique is suggested. For $11 \geq N_\mathrm{c}$, random code generation with hill-climbing procedure is proposed. In this setting, the binary classifier $\gamma^{(j)}$ associated with the j^th column of the coding matrix $\mathscr{C}$ aims at distinguishing the trials which are labeled with 0 from the ones labeled with 1. This results in two super-sets for each classifier $\gamma^{(j)}$ denoted by $\boldsymbol{S}^{(j,0)}$ and $\boldsymbol{S}^{(j,1)}$, with their corresponding features which are extracted to be fed into the classifier. Training an ECOC classifier, therefore, comprises of learning a set $\boldsymbol{\Lambda} = \{\gamma^{(1)}, \dots, \gamma^{(N_\mathrm{Bits})}\}$ of independent binary classifiers. An illustrative example is shown in Figure 3.3 for classifying EEG epochs in to $N_\mathrm{c} = 4$ categories. Figure 3.2 depicts all the associated 16 possible codewords. Hypothesizing an unlabeled EEG epoch $\boldsymbol{X}_i$ is performed based on the set $\boldsymbol{\Lambda}$ of learned classifiers by first evaluating each classifier based on $\boldsymbol{X}_i$ resulting in the generation of the following binary-vector

$$\boldsymbol{\Lambda}(\boldsymbol{X}_i) = \{\gamma^{(1)}(\boldsymbol{X}_i), \dots, \gamma^{(N_\mathrm{Bits})}(\boldsymbol{X}_i)\}. \tag{3.1}$$

Epoch $\boldsymbol{X}_i$ is assigned to a given class if the resulting bit-vector computed from Eq. (3.1) is similar to the codeword representing that class. In the likely case that the resulting bit-vector is not the same as any of the rows of the codebook $\mathscr{C}$, trial $\boldsymbol{X}_i$ will be assigned to the closest codeword as follows

$$\Phi(\boldsymbol{X}_i) = \mathrm{argmin}_m \Delta\big(\mathscr{C}_m, \boldsymbol{\Lambda}(\boldsymbol{X}_i)\big), \tag{3.2}$$

Algorithm 1 ECOC BASED EEG CLASSIFICATION

Input: $\{(\boldsymbol{X}_i, \Omega_i)\}_{i=1}^{N_{\text{Trial}}}$, and; N_{Bits}.
Output: N_{Bits} trained binary classifiers $\{\gamma^{(1)}, \dots, \gamma^{(N_{\text{Bits}})}\}$.
1: *Codebook Generation*: Form ($N_{\text{c}} \times N_{\text{Bits}}$) coding matrix $\mathscr{C}$.
2: *Classifier Design*:
3: **for** $j \in \{1, \dots, N_{\text{Bits}}\}$ **do**
4: Super-sets Construction: $\boldsymbol{S}^{(j,0)} = \{\boldsymbol{X}_i;\ \mathscr{C}_{i,j} = 0\}$ and $\boldsymbol{S}^{(j,1)} = \{\boldsymbol{X}_i;\ \mathscr{C}_{i,j} = 1\}$.
5: Binary Classifier Construction: Train classifier $\gamma^{(j)}$ discriminate between $\boldsymbol{S}^{(j,0)}$ and $\boldsymbol{S}^{(j,1)}$.
6: **end for**

where $\Delta(\boldsymbol{a}, \boldsymbol{b})$ is the number of bits in which vectors a and b differ. More specifically, first the Hamming distance between all the available codewords $\mathscr{C}_i$ and the computed bit-vector $\boldsymbol{\Lambda}(\boldsymbol{X}_i)$ given by Eq. (3.1) is computed. Then epoch $\boldsymbol{X}_i$ is assigned to class n which has the minimum distance among the other codewords. Before presenting the ECCSP, below we present an example to better clarify the ECOC technique.

The ECOC-based CSP Spatial Filtering

The proposed ECCSP framework couples ECOC classification approach presented in Section 3.2.1.B with CSP approach [33]. The latter forms a linear transformation matrix for projecting the EEG epochs into a lower-dimensional spatial subspace, which is then used by the former to construct N_{Bits} classifiers to learn the designed codebook $\mathscr{C}$. Essentially, the CSP approach simultaneously diagonalizes the normalized spatial covariance matrix of EEG epochs associated to the two super-sets to forms projection matrix that maximizes the variance of the two super-sets used by each of the N_{Bits} ECOC classifiers. Intuitively speaking, the reason that the CSP approach outperforms some well regarded analytical techniques (such as Independent Component Analysis (ICA) or the Principal Component Analysis (PCA)) for dimension reduction and feature extraction can be attributed to the fact that the CSP technique uses the labels of the data and handles the problem in a supervised fashion. Features required for classification/training of N_{Bits} binary classifiers are obtained by analyzing the spatially filtered signals.

Each classifier $\gamma^{(j)}$, for ($1 \leq j \leq N_{\text{Bits}}$), within the ECCSP framework, uses its classifier-specific features. More specifically, for trial $\boldsymbol{X}_i$, at first the normalized spatial covariance matrix is computed as follows

$$\boldsymbol{\Sigma}_i = \frac{1}{N_{\text{t}} - 1} \big(\boldsymbol{X}_i - \boldsymbol{\mu}_i\big)\big(\boldsymbol{X}_i - \boldsymbol{\mu}_i\big)^T, \tag{3.3}$$

where $\boldsymbol{\mu}_i$ is the column-wise mean of $\boldsymbol{X}_i$. Typically, EEG signals are bandpassed resulting in the removal of the mean component, i.e., $\boldsymbol{\mu}_i = \boldsymbol{0}$. The full second-order statistical characteristics of the EEG epochs can then be inferred from

$$\boldsymbol{\Sigma}_i = \frac{\boldsymbol{X}_i \boldsymbol{X}_i^T}{\text{Tr}(\boldsymbol{X}_i \boldsymbol{X}_i^T)}. \tag{3.4}$$

To train the classifier $\gamma^{(j)}$, for ($1 \leq j \leq N_{\text{Bits}}$), first its two corresponding super-sets $\boldsymbol{S}^{(j,0)}$ and $\boldsymbol{S}^{(j,1)}$ are formed, and then $\bar{\Sigma}^{(j,0)}$ and $\bar{\Sigma}^{(j,1)}$ are computed by taking the average of all spatial covariance matrices computed based on Eq. (3.4) from all the epochs that belong to the super-sets ($\boldsymbol{S}^{(j,0)}$ and $\boldsymbol{S}^{(j,1)}$), respectively. The next step is to compute the composite spatial covariance matrix $\Sigma^{(j,c)} = \bar{\Sigma}^{(j,0)} + \bar{\Sigma}^{(j,1)}$ and compute its eigenvalue decomposition as follows

$$\Sigma^{(j,c)} = \boldsymbol{V}^{(j,c)} \boldsymbol{\lambda}^{(j,c)} [\boldsymbol{V}^{(j,c)}]^T, \tag{3.5}$$

where matrix $\boldsymbol{V}^{(j,c)}$ is constructed based on the eigenvectors of $\Sigma^{(j,c)}$, and the diagonal matrix $\boldsymbol{\lambda}^{(j,c)}$ is constructed from eigenvalues of the composite matrix $\Sigma^{(j,c)}$. These two terms are then used within the ECCSP framework to form the following whitening transformation matrix $\boldsymbol{P}^{(j)} = \sqrt{\left[\boldsymbol{\lambda}^{(j,c)}\right]^{-1}} [\boldsymbol{V}^{(j,c)}]^T$, which equalizes the variance in the space spanned by the composite matrix's eigenvectors resulting in the eigenvalues of $\boldsymbol{P}^{(j)} \Sigma^{(j,c)} [\boldsymbol{P}^{(j)}]^T$ to become one. The whitening matrix $\boldsymbol{P}^{(j)}$ is then used to transform matrices $\bar{\Sigma}^{(j,0)}$ and $\bar{\Sigma}^{(j,1)}$ (the average covariance matrices associated with each of the two super-sets) as

$$\boldsymbol{S}^{(j,1)} = \boldsymbol{P}^{(j)} \bar{\Sigma}^{(j,1)} [\boldsymbol{P}^{(j)}]^T \text{and } \boldsymbol{S}^{(j,0)} = \boldsymbol{P}^{(j)} \bar{\Sigma}^{(j,0)} [\boldsymbol{P}^{(j)}]^T. \tag{3.6}$$

The transformation introduced in Eq. (3.6) results in matrices $\boldsymbol{S}^{(j,0)}$ and $\boldsymbol{S}^{(j,1)}$ to have common eigenvectors given by $\boldsymbol{S}^{(j,1)} = \boldsymbol{B}^{(j)} \boldsymbol{\lambda}^{(j,1)} [\boldsymbol{B}^{(j)}]^T$ and $\boldsymbol{S}^{(j,0)} = \boldsymbol{B}^{(j)} \boldsymbol{\lambda}^{(j,0)} [\boldsymbol{B}^{(j)}]^T$, where the sum of the two eigenvalues equals an identity matrix $\boldsymbol{I}$ of appropriate dimension (i.e., $\boldsymbol{\lambda}^{(j,0)} + \boldsymbol{\lambda}^{(j,1)} = \boldsymbol{I}$). This neat property results in the eigenvector associated with the largest eigenvalue of $\bar{\boldsymbol{S}}^{(j,0)}$ to have the smallest eigenvalue associated with $\bar{\boldsymbol{S}}^{(j,1)}$ making the eigenvectors $\boldsymbol{B}^{(j)}$ appropriate for classifying the two super-sets. The whitened EEG epochs are then projected onto the eigenvectors associated with the largest and smallest eigenvalues of matrix $\boldsymbol{B}^{(j)}$ to form the optimal feature vectors (in the least square sense) to discriminate the two super-sets. The projection matrix associated with classifier $\gamma^{(j)}$ is, therefore, computed as follows $\boldsymbol{W}^{(j)} = \left[\boldsymbol{P}^{(j)}\right]^T \boldsymbol{B}^{(j)}$, which is finally used to decompose each epoch $\boldsymbol{X}_i$ as

$$\mathscr{W}_i^{(j)} = \left[\boldsymbol{W}^{(j)}\right]^T \boldsymbol{X}_i. \tag{3.7}$$

The j^{th} classifier then constructs its feature vector $\boldsymbol{f}_i^{(j)}$ from the i^{th} EEG trial $\boldsymbol{X}_i$ by only using the first and last m rows of $\mathscr{W}_i^{(j)}$ [33] as follows

$$\boldsymbol{f}_i^{(j)} = \log \left(\frac{\operatorname{var}\left(\mathscr{W}_{i,p}^{(j)}\right)}{\sum_{k=1}^{2m} \operatorname{var}\left(\mathscr{W}_{k,p}^{(j)}\right)} \right), \tag{3.8}$$

where matrix $\mathscr{W}_{i,p}^{(j)}$ contains the first and last m rows of matrix $\mathscr{W}_i^{(j)}$ and $\operatorname{var}(\cdot)$ denotes the variance operator. This completes the introduction of the proposed ECCSP framework summarized in Algorithm 2.

Algorithm 2 TRAINING IN THE ECCSP FRAMEWORK

Input: Training EEG Trials $\boldsymbol{X}$; The coding matrix: $\mathscr{C}$.
Output: N_{Bits} trained binary classifiers $\{\gamma^{(1)}, \ldots, \gamma^{(N_{\text{Bits}})}\}$.
1: According to the coding matrix $\mathscr{C}$, construct the two super-sets for each classifier.
2: Perform spatial filtering on the recordings by means of Eq. (3.7).
3: Select the first and last m rows of the spatially filtered signals.
4: Extract classifier-specific features using Eq. (3.8).

3.2.2 The ECCSP-TB

In this section, we propose a modified variant of the ECCSP, referred to as the ECCSP-TB, which deploys a Bayesian framework to optimize the spectral filters within the ECCSP framework resulting in a set of class discriminative frequency bands and their associated spatial filters. The ECCSP-TB variant is built upon the BSSFO technique proposed in Reference [59]. The ECCSP-TB extends the spatial filtering of the ECCSP by utilization of the BSSFO technique and also extends the multi-class classification approach by using a ternary version of the ECOC classifier. In the following sub-sections, the details of the ECCSP-TB variant are presented.

Bayesian Spatio-spectral Filter Optimization

The ECCSP-TB performs spectral filtering and spatial filtering on the EEG signals as described below.

- **Spectral Filtering:** This step is modeled as the convolution of the input EEG epoch $\boldsymbol{X}_i$ with the impulse response of a bandpass filter $h(k)$, i.e.,

$$\mathscr{Z}_i = h \circledast \boldsymbol{X}_i, \tag{3.9}$$

where $\circledast$ denotes the convolution operator, and the cutoff frequency vector $\mathscr{B}$ specifying the bandpass filter h is defined as $\mathscr{B} \triangleq [b_s, b_e]^T$. The cutoff frequencies are modeled as random variables and are optimized iteratively to increase the classification accuracy. In other words, the uncertainty in the cut-off frequencies of the spectral filters are modeled with a prior probability denoted by $p(\mathscr{B})$ over random variable $\mathscr{B}$. The prior density $p(\mathscr{B})$ describes relative probabilities of different states (frequency bands) in which a single-trial EEG recording is correctly discriminated. The posterior probability distribution is then computed based on each bandpass-filtered single-trial EEG recording $\mathscr{Z}_i$, for $(1 \leq i \leq N_{\text{Trial}})$, and its corresponding label denoted by Ω_i, as

$$p(\mathscr{B}|\mathscr{Z}_i, \Omega_i) = \frac{p(\mathscr{Z}_i, \Omega_i|\mathscr{B})p(\mathscr{B})}{p(\mathscr{Z}_i, \Omega_i)}. \tag{3.10}$$

However, term $p(\mathscr{Z}_i, \Omega_i|\mathscr{B})$ on the Right-hand Side (RHS) of Eq. (3.10) is too complex in nature resulting in complex $p(\mathscr{B}|\boldsymbol{X}_i,\Omega_i)$, which eliminates the possibility of direct evaluation in closed-form. Alternatively, particle-based approximation techniques [220] are utilized to address this issue. In brief, a set of N_p particles $\{\mathbb{B}(k)\}_{k=1}^{N_p}$ generated from the prior density $p(\mathscr{B})$ are utilized, where $\mathbb{B}(k)$ denotes a particle representing a single filter bank. To be more specific, each particle set $\mathbb{B}(k)=\{b_s(k),b_e(k),\pi(k)\}$ contains the characteristics of the spectral filter $(b_s(k),b_e(k))$ together with its associated weight $(\pi(k))$. Hence, the particle-based posterior distribution (Eq. (3.10)) can be written as

$$p(\mathbb{B}(k)|\mathscr{X}_i(k),\Omega_i)=\frac{p(\mathscr{X}_i(k),\Omega_i|\mathbb{B}(k))p(\mathbb{B}(k))}{p(\mathscr{X}_i(k),\Omega_i)}. \tag{3.11}$$

The posterior $p(\mathbb{B}(k)|\mathscr{X}_i(k),\Omega_i)$ provides all the required information regarding $\mathscr{B}$ which can be obtained from the bandpass-filtered signal $\mathscr{X}_i(k)$ and its corresponding class label Ω_i. The spectral filtering step is then followed by computing the common spatial patterns of each trial in each particle (i.e., $\mathscr{X}_i(k)$).

- **Spatial Filtering:** This step is performed using the CSP method (Eq. (3.7)). However, since particles are deployed to solve the optimization problem, the spatial filtering is coupled with spectral filtering, i.e., Eq. (3.7) is modified as

$$\mathscr{W}_i^{(j)}(k)=\left[\boldsymbol{W}^{(j)}(k)\right]^T\mathscr{X}_i(k). \tag{3.12}$$

Please note that from this stage, the superscript "$(.)^{(j)}$" $(1\leq j\leq N_{\text{Bits}})$ is added to the equations to demonstrate that the ECOC formulations are now in effect. Eq. (3.12) performs the spatial filtering of the spectrally filtered signals and prepares them for the feature extraction step which is introduced in Eq. (3.8). After calculating the features for each trial based on the characteristics of each particle, a feature matrix containing all the features for all the trials is formed as follows

$$\boldsymbol{F}^{(j)}(k)=\{\boldsymbol{f}_{i,p}^{(j)}(k)\}_{i=1}^{N_{\text{Trial}}}\in\mathbb{R}^{2m\times N_{\text{Trial}}}. \tag{3.13}$$

Since the spectro-spatial filtering is a deterministic procedure and does not add any stochastic behavior to the system, the posterior probability in Eq. (3.11) can be rewritten as

$$p(\mathbb{B}(k)|\mathscr{X}_i(k),\Omega_i)\triangleq p(\mathbb{B}(k)|\boldsymbol{F}^{(j)}(k),\boldsymbol{\Omega})=\frac{p(\boldsymbol{F}^{(j)}(k),\boldsymbol{\Omega}|\mathbb{B})p(\mathbb{B}(k))}{p(\boldsymbol{F}^{(j)}(k),\boldsymbol{\Omega})}. \tag{3.14}$$

In order to compute the likelihood probability $p(\boldsymbol{F}^{(j)}(k),\boldsymbol{\Omega}|\mathbb{B})$ in Eq. (3.14), the mutual information [221] is utilized to compute the likelihood function and measure the discriminative power of the features in classifying the EEG trials. In this regard, the likelihood is defined as

$$p(\boldsymbol{F}^{(j)}(k),\boldsymbol{\Omega}|\mathbb{B})=\exp\left[I\big(\boldsymbol{F}^{(j)}(k);\boldsymbol{\Omega}\big)\right], \tag{3.15}$$

where $I(.)$ denotes the mutual information between the features matrix and the class labels. Please note that the mutual information is actually calculated between the columns of the features matrix (i.e., each feature vector) and the class labels, therefore, it can reveal how much each feature vector is representing the class labels. The mutual information is computed based on the entropy of the features and the labels, i.e.,

$$I(\boldsymbol{F}^{(j)}(k);\boldsymbol{\Omega}) = H(\boldsymbol{F}^{(j)}(k)) - H(\boldsymbol{F}^{(j)}(k)|\boldsymbol{\Omega}), \tag{3.16}$$

where $H(.)$ and $H(.|.)$ represent the entropy and conditional entropy, respectively [221]. Once the posterior probabilities are estimated, the weights of the particles need to be derived. The weight of each particle is computed as follows

$$\pi^{(j)}(k) = \frac{p(\boldsymbol{F}^{(j)}(k), \boldsymbol{\Omega}|\mathbb{B}_k)}{\sum_{k=1}^{N_p} p(\boldsymbol{F}^{(j)}(k), \boldsymbol{\Omega}|\mathbb{B}(k))}. \tag{3.17}$$

For each iteration of the optimization procedure, the set of particle weights are calculated according to Eq. (3.17). Then, the set $\boldsymbol{\Pi}^{(j)}$ is formed according to the following criteria

$$\boldsymbol{\Pi}^{(j)} = \bigcup_k (\pi^{(j)}(k) > \tau). \tag{3.18}$$

The criteria introduced in Eq. (3.18) selects the particles with weights ($\pi^{(j)}(k)$) greater than a given threshold τ, which is a random number between 0 and 1 changing at each iteration. If a particle is selected and is included in the set $\boldsymbol{\Pi}^{(j)}$, its associated characteristics of the filter bank remain intact for the next iteration. However, for those particles which are not included in this set, a disturbance following a normal distribution ($\sim \mathcal{N}(0,1)$) is added to the values of the band limits, which define the characteristics of the filter banks. In the next iteration, the effect of the disturbance (changing the characteristics of the filter banks) is evaluated, and the same procedure goes on for a specific number of iterations until the particles converge to the optimum value of the band limits for the spectral filter and the particle weights. In the final step, N_p number of SVM classifiers are trained based on the features extracted from each article.

In the evaluation phase, since N_p number of particles are optimized, N_p different spectral filters are applied to the trials and the process is followed by applying spatial filters corresponding to each particle. Then the features of each particle are obtained and are fed into the corresponding classifier. To merge the decisions of N_p classifiers, the output of each classifier ($+1$ or -1) is multiplied by the weight of the particle and the final decision of the ECCSP-TB is based on the weighted summation of the outputs of the classifiers. Please note that in order to obtain a smooth classification scheme in the evaluation part, the two classes of data are labeled with $+1$ and -1. As stated above, this labeling comes in handy for merging the decision of the classifiers by incorporating the weight of the particles. Finally, the classifiers' set for all particles is denoted as $\{\zeta(k)\}_{k=1}^{N_p}$, therefore, the decision rule in the ECCSP-TB is defined as

$$\lambda_{test}^{(j)} = \text{sign}\big(\sum_{k=1}^{N_p} \pi^{(j)}(k)\zeta(k)(\boldsymbol{X}_{test})\big). \tag{3.19}$$

Next, the modification of the classification component in comparison to the primary ECCSP framework is presented.

Ternary ECOC Classification Scheme

As stated previously, the ECOC technique splits the problem of multi-class classification into a number of binary sub-classification problems. The action of how to split the problem, requires generating proper codewords. In the previous sections, the deployment of binary codewords is being discussed but here we propose to extend the ECCSP by generating codewords based on the ternary numerical system.

We believe that the drawback of the ECOC method, introduced in Section 3.2.1, is that in each of the cases that the super-sets ($S^{(j,0)}$ and $S^{(j,1)}$) are formed, all the classes are involved and there is no chance to examine the classifier $\gamma^{(j)}$ in absence of features of a certain class. Therefore, we propose the Ternary ECOC classifiers, where the possible labels are now 0, 1 and 2 (ternary system). In this work, 0 and 1 are taken as labels to form the super-sets, and the classes with label 2, are not involved in the classifier design. By performing a similar procedure to that of the conventional ECOC technique for removing the complements and meaningless cases, and taking this fact into account that label “2" means that the class is not involved into the classifier training, 25 particular cases are derived. It is also worth mentioning that any 4-digit ternary number which does not consist of either 0 or 1, is defined as meaningless. Table 3.1 demonstrates the ternary codewords which are derived for classification of 4 classes of data. If proper original codewords are assigned to the classes with enough Hamming distance, the misclassification error of the binary classifiers can be handled, up to some extent, and the correct class codeword could be predicted. This is the main difference between the ECOC classifiers and the OVO and OVA techniques. The benefit of the proposed TECOC classification scheme over its conventional counterpart is introducing class codewords that have higher Hamming distance and, therefore, are capable of handling more misclassification errors. On the other hand, a higher number of trained classifiers and increased computational cost are the drawbacks of this technique. The decision rule for Ternary ECOC is the same as the one in Eq. (3.2), keeping in mind that the code words here are the ternary ones. Algorithm 3 outlines the training phase of the ECCSP-TB framework. Algorithm 4 demonstrates the work-flow in the evaluation phase.

3.2.3 The ECCSP-TB2B

In the previous section, we developed the ECCSP-TB variant, which aims at deriving the optimized spectral filters to better discriminate the EEG trials for different tasks. In the simulations results, its superior performance among different competing techniques was showed. In order to further boost the performance of the proposed ECCSP framework,

Table 3.1: Generated ternary codewords for classifying 4 classes of data in TECOC approach.

Class Label	**Assigned class codeword in ternary system**
Tongue MI	1012101201201210121101211
Foot MI	0111200011122201112011120
Left hand MI	0000000000000000000222222
Right hand MI	0000011111111122222000002

Algorithm 3 THE ECCSP-TB IN TRAINING PHASE

Input: EEG recordings $\{\boldsymbol{X}_i\}_{i=1}^{N_{\text{Trial}}}$ and their corresponding labels $\{\Omega_i\}_{i=1}^{N_{\text{Trial}}}$

Output: N_p number of optimized particles $\{\mathbb{B}(k)\}_{k=1}^{N_p}$ for each bit of (N_{Bits} in total) the ECOC classifier.

for The number of bits in the ECOC codeword (N_{Bits}) **do**
  Construct the two super-sets $\boldsymbol{S}^{(j,0)}$ and $\boldsymbol{S}^{(j,1)}$
  Define N_p number of particles which $\forall k : \pi(k) = \frac{1}{N_p}$.
  Randomly initialize the filter bank band limits.
  for The number of iterations **do**
    for $k \in$ the number of particles **do**
      Spectrally filter the signals by Eq. (3.9)
      Derive the CSP filters for each frequency band
      Spatially filter the signals by Eq. (3.7)
      Extract features by Eq. (3.8)
    end for
    Form the features matrix $\boldsymbol{F}$ by Eq. (3.13)
    Compute the posterior probability for Eq. (3.14)
    Calculate the particle weights using Eq. (3.17)
    if $\pi(k) < \tau$ **then**
      Add normal noise to $[b_s, b_e]$ in particle k.
    end if
  end for
  Train N_p number of SVM classifiers based on the features matrix $\boldsymbol{F}(k)$, from the last iteration.
end for

in this section, we develop the ECCSP-TB2B variant. Here, the Bayesian framework developed in Section 3.2.2 is incorporated to optimize the characteristics of a filter bank (2 frequency bands), instead of a single spectral filter used in the ECCSP-TB variant. The motivation for this modification is to combine the benefits of the filter bank method and the optimized methods, in a way to reach higher accuracies. The reason for selecting two frequency bands is that we believe that the motor related features of the EEG signals are mainly stored in the μ band ($8-13$Hz) and the β band ($13-30$Hz). Ideally speaking, it is expected that the limits of the incorporated filter bank converge such that the optimized limits maximize the useful extracted frequency contents of the aforementioned bands. In this regard, a filter bank consisting of 2 bandpass filters is initialized, and then the frequency limits are optimized based on the procedure discussed in Section 3.2.2. More specifically, In the ECCSP-TB2B variant, the randomness of the cut-off frequencies for the filter bank are modeled with *a priori* probability denoted by $p(\mathscr{B}^{\mathscr{D}})$ over random variable $\mathscr{B}^{\mathscr{D}}$. In this scheme, three random variables are deployed to characterize the filter bank and $\mathscr{B}^{\mathscr{D}}$ is defined as $\mathscr{B}^{\mathscr{D}} \triangleq [b_s, b_m, b_e]$, where ideally the μ band contents are to be extracted with the $[b_s, b_m]$ bandpass filter and the β band contents to be extracted with the $[b_m, b_e]$ bandpass filter. The idea of considering a same cut-off-frequency b_m for both bandpass filters mainly relies on the typical values of the frequency bands that are reported in physiology, i.e., the μ and the β bands have 13 Hz in common. Similar to the ECCSP-TB variant, the posterior density denoted now by $p(\mathscr{B}^{\mathscr{D}}|\boldsymbol{X}_i, \Omega_i)$ is constructed from single-trial EEG recording $\boldsymbol{X}_i$, for $(1 \leq i \leq N_{\text{Trial}})$, and its particle-based approximation is

$$p(\mathbb{B}^{\mathscr{D}}(k)|\boldsymbol{X}_i, \Omega_i) = \frac{p(\boldsymbol{X}_i, \Omega_i|\mathbb{B}^{\mathscr{D}}(k))p(\mathbb{B}^{\mathscr{D}}(k))}{p(\boldsymbol{X}_i, \Omega_i)}, \tag{3.20}$$

where $\mathbb{B}^{\mathscr{D}}(k) = \{b_s(k), b_m(k), b_e(k), \pi(k)\}$ denotes particle k, for $(1 \leq k \leq N_p)$, and represents a single filter bank. The bandpass filtering of the signals is modeled as a convolution operation on the input signals performed by the system $h_l(k)$, for $(l \in \{1,2\})$, associated with each of the two bandpass filters. Please note that $l = 1$ determines the filter for $[b_s, b_m]$, and $l = 2$ denotes the filter for $[b_m, b_e]$. Therefore, the filtered signal, denoted by $\mathscr{Z}_l$, is deterministically obtained as

$$\mathscr{Z}_{i,l}(k) = h_l(k) \circledast \boldsymbol{X}_i, \tag{3.21}$$

The likelihood and the evidence are, therefore, become equal to $p(\mathscr{Z}_{i,l}(k), \Omega_i|\mathbb{B}^{\mathscr{D}}(k))$ and $p(\mathscr{Z}_{i,l}(k), \Omega_i)$, respectively. Hence, Eq. (3.20) can be rewritten by replacing the raw EEG signal $\boldsymbol{X}_i$ with its bandpass-filtered version $\mathscr{Z}_{i,l}(k)$ as

$$p(\mathbb{B}^{\mathscr{D}}(k)|\mathscr{Z}_{i,l}(k), \Omega_i) = \frac{p(\mathscr{Z}_{i,l}(k), \Omega_i|\mathbb{B}^{\mathscr{D}}(k))p(\mathbb{B}^{\mathscr{D}}(k))}{p(\mathscr{Z}_{i,l}(k), \Omega_i)}. \tag{3.22}$$

The spectral filtering step is then followed by computing the common spatial patterns of each trial for each frequency band in each particle (i.e., $\mathscr{Z}_{i,l}(k)$). The spatial filtering formula is rewritten as $\mathscr{W}_{i,l}^{(j)}(k) = \left[\boldsymbol{W}_l^{(j)}(k)\right]^T \boldsymbol{Z}_{i,l}(k)$, and the feature extraction formula is updated as

$$\boldsymbol{f}_{i,l}^{(j)}(k) = \log\left(\frac{\text{var}\left(\mathscr{W}_{i,l,p}^{(j)}(k)\right)}{\sum\limits_i \text{var}\left(\mathscr{W}_{i,l,p}^{(j)}(k)\right)}\right). \tag{3.23}$$

Now, at this stage, the features of the two parallel pipelines are merged into one single feature vector as

$$\boldsymbol{f}_i^{(j)}(k) = \left[f_{i,l}^{(j)}(k)|_{l=1}, \boldsymbol{f}_{i,l}^{(j)}(k)|_{l=2}\right], \tag{3.24}$$

Algorithm 4 THE ECCSP-TB IN EVALUATION PHASE

Input: Unlabeled EEG Trial $\boldsymbol{X}_{\text{Test}}$, N_p number of trained binary classifiers, optimized particles $\{\mathbb{B}(k)\}_{k=1}^{N_p}$ and spatial filters $\boldsymbol{W}_l(k)$.

Output: The predicted class label for unseen trial $\boldsymbol{X}_{\text{Test}}$.

1: **for** The number of particles N_p **do**

2: Spectral filtering of the trial based on Eq. (3.9).

3: Spatial filtering of the trials according to Eq. (3.7).

4: Extract classifier-specific features using Eq. (3.8).

5: Obtain the prediction of each classifier for each set of features, λ .

6: **end for**

7: Compute the weighted summation of the predictions.

8: Obtain the final decision based on the sign of the weighted summation.

and then the vectors are combined according to Eq. (3.13). The rest of the procedure is the same as discussed in Section 3.2.2.

3.2.4 Simulation Results

In this section, an extensive set of simulation experiments to evaluate the performance of the proposed ECCSP framework and its variants are presented. The simulations are performed based on two well regarded MI datasets provided by BCI Competition IV, referred to as "BCIC-IV_{2a}" and "BCIC-IV_{2b}" [218].

- Dataset BCIC-IV_{2a}: This dataset consists of four MI classes of EEG measurements (Right hand MI, Left hand MI, Foot MI, and Tongue MI), obtained from 9 healthy subjects. Signals are recorded with sampling rate of 250Hz, using 22 EEG channels and 3 monopolar Electrooculogram (EOG) channels (with left mastoid serving as the reference). The original EEG signal recordings are already bandpass filtered ($0.5-100$Hz) and notch filtered to remove the interference of the power line on the signals. For each subject, two sessions are recorded (one for training purposes and the other one for evaluation). Each session consists of six segments, and each segment includes 48 trials of length 3 seconds. In total and for each subject, 288 trials for training and 288 trials for evaluation are available. In order to measure the performance of the proposed ECCSP framework and based on the recommendation from BCI competition [218], kappa coefficient κ is used, i.e., $\kappa = \frac{\text{CCR}-P_{\text{rand}}}{1-P_{\text{rand}}}$, where CCR represents the correct classification rate, and the value of $P_{rand} = 0.25$. All the algorithms are implemented only based on the 22 EEG channel recordings.
- Dataset BCIC-IV_{2b}: This dataset consists of the EEG recordings from 9 different right-handed subjects, with normal or corrected-to-normal vision. The subjects were asked to perform two MI tasks including "Left Hand MI" and "Right Hand MI." From each subject, 5 sessions are recorded, 3 of them for training and the rest for evaluation. Among the training sessions, 2 of them are recorded without providing feedback to the subject, and one is recorded when the feedback is enabled. This dataset

Table 3.2: Performance comparison for different approaches, tested on BCIC-IV_{2b} dataset. Performance measure is in Kappa (κ) value.

Subjects	**ECCSP**	**ECCSP-TB** (BSSFO)	**FBCSP**	**ECCSP-TB2B**
Subject 1	0.15	0.19	0.21	**0.28**
Subject 2	0.01	0.10	0.16	**0.17**
Subject 3	**0.24**	0.07	-0.04	0.09
Subject 4	0.37	0.92	0.61	**0.96**
Subject 5	0.24	0.20	0.55	**0.58**
Subject 6	-0.05	0.45	0.23	**0.61**
Subject 7	0.44	0.45	0.31	**0.56**
Subject 8	0.74	0.77	0.18	**0.80**
Subject 9	0.31	0.62	0.13	**0.64**
Average	0.27	0.42	0.26	**0.52**

Table 3.3: Performance comparison in Kappa values (κ) for different approaches, tested on $BCIC-IV_{2a}$ for prediction on test data.

Subjects	**ECCSP**	**SCSSP** [57]	**FBCSP** [57]	**FBCSP** [54]	**BSSFO OVO**	**BSSFO OVA**	**TSSM+SVM** [53]	**BSSFO Conventional ECOC**	**ECCSP-TB**	**ECCSP-TB2B**
Subject 1	0.48	0.62	0.68	0.68	0.62	0.31	**0.77**	0.52	0.60	0.75
Subject 2	0.20	0.27	0.30	0.42	0.20	0.08	0.33	0.14	0.25	**0.44**
Subject 3	0.49	0.66	0.71	0.75	0.70	0.57	0.77	0.56	0.64	**0.80**
Subject 4	0.31	0.27	0.39	0.48	0.41	0.45	0.51	**0.54**	0.50	**0.54**
Subject 5	0.16	0.07	0.28	**0.40**	0.09	-0.07	0.35	0.07	0.15	0.27
Subject 6	0.13	0.26	0.25	0.27	0.15	0.08	**0.36**	0.18	0.24	0.31
Subject 7	0.64	0.41	0.57	0.77	0.72	0.62	0.71	0.64	0.75	**0.80**
Subject 8	0.38	0.59	0.59	**0.76**	0.58	0.30	0.72	0.52	0.64	0.73
Subject 9	0.38	0.66	0.54	0.61	0.65	0.55	**0.83**	0.56	0.69	0.75
Average	0.35	0.42	0.48	0.57	0.46	0.32	0.59	0.41	0.50	**0.60**

provides 6 recording channels, 3 for EEG and 3 for EOG. The EEG channels are recorded form C3, Cz, and C4 points in the 10-20 EEG recording system, similar to the previous dataset, only the EEG recordings are used. The signals are recorded with sampling frequency of 250Hz and bandpass filtered between $0.5-100$Hz. A notch filter is also applied to remove the 50Hz effect of the power line on the recordings. For this dataset, the performance measure is also kappa coefficient where $P_{rand} = 0.5$.

Parameter Selection

As stated previously, there are two hyper-parameters in the proposed framework to be optimized, i.e., the number of iterations and the number of particles. As simultaneous optimization of these parameters is time-consuming and computationally expensive, a grid search approach is employed for optimizing these two parameters in an independent fashion. The assumption of taking the parameters independent of each other is common in multi-parameter phenomena, such as naive Bayes classifiers. For each of the parameters, values from 5 to 50 with the step size of 5 are considered and the performance of the framework is evaluated through 10×10-fold cross-validation. The parameter selection is

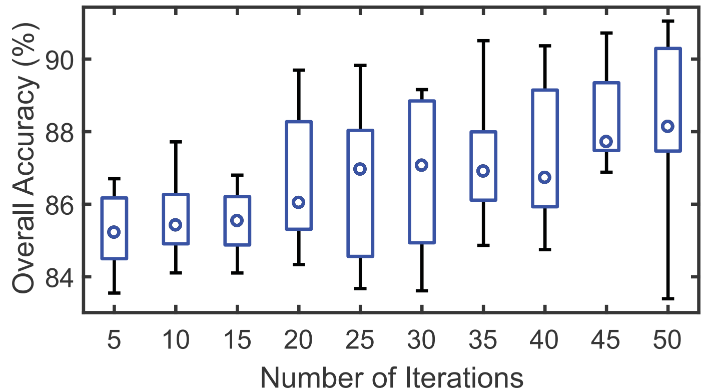


Figure 3.4: 10×10-fold cross-validation with different number of iterations: 45 iterations are selected as the optimum one.

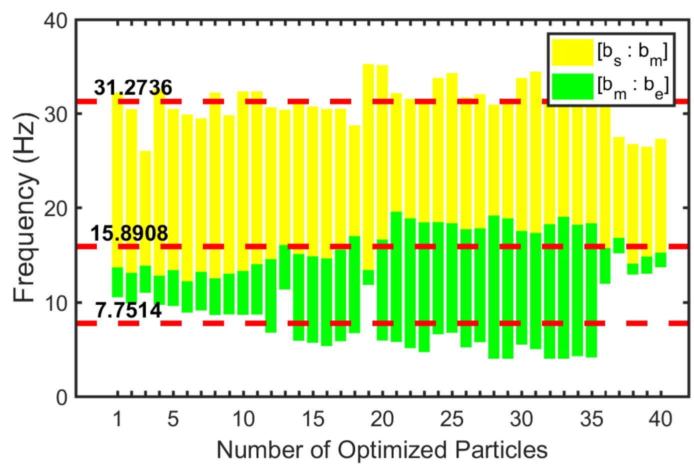


Figure 3.5: Optimized frequency bands for all of the particles within 45 iterations.

performed for each subject individually and subject-specific parameters are then employed for computing the final accuracies. We emphasize that only the training samples are used for the parameter selection process and test samples are not used within this process.

- **Number of Iterations:** In this experiment, the number of iterations to achieve the maximum cross-validation performance is investigated. To this aim, the number of particles is kept constant equal to 30. This number is reported by [59], which was the first paper proposing the utilization of a Bayesian framework for spatial-spectral filter optimization. Then the number of iterations is varied from 5 to 50 with steps of 5, and the 10×10-fold cross-validation of the performance is measured. The results are presented in Figure 3.4. It is observed that the number of iterations providing the best performance in terms of maximum mean and minimum variance is 45. Please note that, although for 50 number of iterations higher mean performance is achieved, the variance the range of results is wider making it undesirable.

- **Number of Particles:** Here, the number of iterations is set to 45 and the number of particles are varied within the range of 5 to 50 with steps of 5 to select the best option for the number of particles. Likewise the previous experiment, 10×10-fold cross-validation is deployed to measure the performance of the system. However, in contrary to the previous experiment, the number of particles are optimized separately for each individual, i.e., a subject-specific hyper-parameter selection procedure.

Experimental Procedure, Results, and Discussions

Segments of 2 seconds (500 samples) from each trial are selected as the input to each framework. The segment starts 0.5 seconds after the onset cue of each trial and lasts for 2 seconds. For the spectral filters which are used in all of the algorithms, a 5^{th} order Butterworth bandpass filter is deployed. Since the optimization procedure is time-consuming and computationally extensive, we decided to deploy a low order spectral filter but kept it the same across all the algorithms for fair comparison. For the spatial filtering of the signals, the CSP technique in its most basic form is utilized. The reason for this selection is that our main objective is to demonstrate the effectiveness of deploying optimized spectral filter banks on the overall performance of the system, therefore, basic forms of the spatial filtering and feature extraction are used. After the spectral filtering step, the signals are fed to the CSP method to perform dimensionality reduction and spatial filtering. To identify the optimal number of new dimensions, for dataset IV_{2a}, 10-fold cross-validation is performed on the training samples and $m = 2$ is selected. For dataset IV_{2b}, as only three recording channels are available, $m = 1$ is selected for dimensionality reduction process. In the classification step, the SVM is used as the underlying binary classifier of the ECOC classifiers. It is worth mentioning that the linear SVM classifier is deployed. In the training phase and according to the procedure described in Section 3.2.1, the generated codewords for the 4 class classification problem are employed, which are introduced in Figure 3.2 and Table 3.1. When it comes to the evaluation phase, the extracted test trials are fed into each trained particle and are filtered according to the specific spectral characteristics of that particle. The votes of the particles are then integrated into the final decision based on Eq. (3.19). This whole process produces one bit out of the N_{Bits} required for the Ternary ECOC classifier to produce the final decision based on Eq. (3.2).

Results from eight different approaches are computed and compared for accuracy in Table 3.3. In particular, we have implemented and compared the proposed frameworks against the results of the following algorithms: (i) The FBCSP method [54], which is the winner of the BCI competition *IV* and achieved the highest performance for this dataset; (ii) The SC-SSP method [57], which is one of the most recent works on this problem; (iii) The FBCSP results reported in Reference [57], which are computed without incorporation of the mutual information-based feature selection step; (iv) The best variant of the Tangent Space of Sub-Manifold (TSSM) approach [53], which is considered as the state-of-the-art in statistical (engineered and not data driven) solutions, and; (v)-(vi) The BSSFO [59] framework when coupled with the proposed Ternary ECOC and conventional ECOC classifiers. In addition, the simulation results for the two techniques, which serve as extensions of binary classifiers for multi-class classification problems are provided. The two techniques are called OVO and OVA, which are extensively used for the multi-class classification problems.

The performance of the proposed frameworks evaluated on dataset BCIC-IV_{2a} is provided in Table 3.3 and the results for dataset BCIC-IV_{2b} is presented in Table 3.2. For the

ECCSP-TB and ECCSP-TB2B variants, it is worth mentioning that in the first iteration, the particles are weighted equally ($1/N_p$), therefore, the initial value for the particle weights is set to $1/40$ and the band limits of the spectral filters are initialized with random numbers within the range of 4-40 Hz.

Please note that for all of the techniques, the spectral filter which is used is a 5^{th} order Butterworth bandpass filter. For the dataset BCIC-IV_{2b}, the proposed ECCSP-TB and ECCSP-TB2B variants are employed without the TECOC classification scheme, as this dataset introduces a binary classification problem. As it is observed from Table 3.3 and Table 3.2, the proposed ECCSP-TB2B framework outperforms its variations as well as other existing frameworks. In particular and in comparison to [53], which is considered as the state-of-the-art in statistical solutions, while the overall result obtained from the proposed framework is slightly better than the best result (TSSM+SVM) reported in [53], my proposed framework outperforms this solution over 5 of the subjects and has less computational complexity during the evaluation phase in comparison to the TSSM+SVM solution.

To better investigate and illustrate the effects of employing personalized spectro-spatial filters, the results of the FBCSP method both with inclusion of the feature selection step [57], and without incorporation of the feature selection procedure are embedded in Table 3.3 and Table 3.2. Therefore, to better track the effects of the optimized filters, the rest of the processing pipeline is kept as simple as possible. The results in the third and forth columns of Table 3.3 are as reported in Reference [57]. Intuitively speaking, the difference between these results with the ones reported in the original paper [218] (the fifth column of Table 3.3) can be attributed to the feature selection technique based on mutual information of the features and the labels. The feature selection step shrinks the dimensionality of the features space from 36 to a new dimension between 4 to 8 and thus a considerable fraction of the features, which are not informative and potentially mislead the classifier are removed. The proposed framework, however, uses only two optimized and subject-specific spectral filters which is computationally far less expensive than the FBCSP and eliminates the need for performing feature selection.

As a final experiment, the cutoff frequencies of the filter bank upon finalizing the optimization process are studied. Figure 3.5 shows the results for subject 1 in dataset IV_{2b}. Figure 3.5 shows the converged frequency bands of the particles used to optimize the subject-specific spectral filters. As the cutoff frequencies of a filter bank are being optimized which consists of two spectral filters, two different colors (green and yellow) are employed. Figure 3.5 is introduced to support the intuition that the proposed framework more closely follows the physiological clues. Sorting the particles in descending order in terms of their associated weights reveals that the first particles have converged to the values that are more compatible with the typical values reported in physiology.

3.3 Siamese Neural Networks for EEG Processing

To develop BCI systems for rehabilitation purposes, accurate identification and extraction of ERD and ERS is crucially important, which explains development of several processing solutions in the literature, including CSP [52] and its extensions [34–37, 54–59]. CSP technique derives a transformation matrix through a supervised process, which not only reduces the dimensionality of EEG signals but also minimizes the variation across each class and maximizes the distance between classes [53]. CSP-based solutions are well suited for binary problems and although they generalize well on small datasets, their classification accuracy drops noticeably with the increase of the number of classes. More recently, successful application of artificial neural network architectures in a variety of domains have ignited a surge of interest in utilization of DL methods for EEG processing applications. Despite the decent performance of existing DL-based methods in development of EEG-based BCI systems [222], the data-hungry nature of DL techniques limits their widespread and reliable application in practical settings.

To tackle the aforementioned issue, a branch in DL techniques is developed, referred to as Siamese Networks [180]. Siamese networks process two inputs in parallel and are specialized in detecting if the two inputs are drawn from the same class or not. The dual input strategy for Siamese networks drastically increases the number of training examples and enables us to take advantage of DL methods for small datasets. Stemmed in their successful application in verification and classification [197, 210] tasks, a growing number of recent works in biomedical domain [222–229] are dedicated to utilization of Siamese networks. In this paper and for the first time in the BCI domain, to the best of our knowledge, employment of Siamese architectures for classification of MI tasks based on EEG signals is proposed. More specifically, the contributions in this section are twofold: (i) Development of an algorithmic procedure to employ Siamese networks for multi-class classification problems, and; (ii) Successfully demonstrating feasibility of using Siamese networks in BCI applications, with the potential of enhancing the classification accuracy.

3.3.1 Siamese Architectures

As discussed in Section 2.2.3 and depicted in Figure 3.6, in Siamese architectures, two identical neural networks with tied parameters are employed to process two input signals in parallel. The outputs are topped with an energy function, which measures the contrast between the two inputs. For the EEG classification problem at hand, each EEG trial is denoted with $\{\boldsymbol{X}_k^{N_{ch}\times N_s}\}_{k=1}^{N_{trials}}$, where N_{ch} and N_s represent the number of channels and the number of samples, respectively. The preprocessing step is modeled as the application of a nonlinear function $f(\cdot)$ on each trial $\boldsymbol{X}_k^{N_{ch}\times N_s}$, resulting in $\boldsymbol{Z}_k = f(\boldsymbol{X}_k)$ as the output of the pre-processing step.

To train and evaluate a Siamese network, in each iteration, the network is fed with $[\boldsymbol{Z}_1, \boldsymbol{Z}_2, Y]$, where $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_2$ are two randomly selected EEG trials and Y is the label denoting if they are from the same class or not. Modeling the effect of network as a function $(G_W(.))$ with parameters W, the Euclidean distance between the outputs of the network for the two inputs is calculated as follows

$$D_W(\boldsymbol{Z}_1, \boldsymbol{Z}_2) = ||G_W(\boldsymbol{Z}_1) - G_W(\boldsymbol{Z}_2)||_2. \tag{3.25}$$

By taking D_W as the short form of $D_W(\boldsymbol{Z}_1, \boldsymbol{Z}_2)$, the loss function for p ($p \leq N_{trials}^2$) number of training pairs is defined as follows

$$l(W) = \sum_{i=1}^{p} L(W, (\boldsymbol{Z}_1, \boldsymbol{Z}_2, Y)^i), \tag{3.26}$$

where $L(W, (\boldsymbol{Z}_1, \boldsymbol{Z}_2, Y)^i) = (1-Y)L_s D_W^i + Y L_D D_W^i$. Terms L_S and L_D indicate the partial loss function for similar pairs and dissimilar pairs, respectively. The loss function is rewritten in the following form

$$L(W, \boldsymbol{Z}_1, \boldsymbol{Z}_2, Y) = (1-Y)\frac{1}{2}D_W^2 + (Y)\frac{1}{2}max(0, m - D_W)^2, \tag{3.27}$$

where $m > 0$ denotes a margin (radius around $G_W(\boldsymbol{Z})$) to decide if a pair of signals are similar or not. In the above formulation, m is a hyper-parameter of the network and W is optimized through the training procedure. The number of training pairs p, could go up to the square of the number of available training samples.

3.3.2 Multi-class Classification Paradigms

As discussed previously, the goal of the feature translation module in a BCI system is to correctly assign the extracted features in previous modules to physiological phenomena. Performance of classifiers normally degrades as the number of studied phenomena increases, which corroborates the urge for development of classifiers with higher learning capacities. Generally speaking, to handle multi-class classification problems, there are two main approaches: (i) Employment of classifiers that are naturally capable of handling multi-class problems, e.g., k-means, k-NN, and Decision Trees to name but a few, and; (ii) Decomposing the multi-class problem into a number of binary classification problems, where binary classifiers could be employed. Due to the widespread utilization of binary feature extraction techniques in the BCI domain, e.g., CSP, the latter technique for multi-class problems is thoroughly investigated. To this aim, two techniques, i.e., OVO and OVR, are typically used in the literature. In both methods, a coding matrix ($\boldsymbol{\Sigma}$) based on binary codewords is employed, which identifies the categorization of training trials into two supersets. Coding matrices for OVO and OVR are presented in Tables 3.4 and 3.5, respectively. The number of columns in $\boldsymbol{\Sigma}$ determines the number of binary classifiers to be trained. For classifier Λ^j, two supersets $\boldsymbol{S}_0^j$ and $\boldsymbol{S}_1^j$ need to be formed, where all the classes denoted by 0 and 1 form supersets $\boldsymbol{S}_0^j$ and $\boldsymbol{S}_1^j$, respectively. Given the coding matrix Σ for a 4-class problem and the set of training trials $\{\boldsymbol{X}_i, Y_i\}_{i=1}^{p}$, where Y_i denotes the trial's label (numbers in the range $[1,2,3,4]$), supersets are formed as follows

$$\begin{cases} \text{if} \quad C^{y,j} = 0: & \boldsymbol{X} \rightarrow \boldsymbol{S}_0^j \\ \text{if} \quad C^{y,j} = 1: & \boldsymbol{X} \rightarrow \boldsymbol{S}_1^j \\ \text{if} \quad C^{y,j} = 2: & \text{No action.} \end{cases} \tag{3.28}$$

In Tables 3.4 and 3.5, the length of each codeword identifies the number of binary classification problems to be solved. As the proposed Siamese architecture provides a binary

Table 3.4: OVO coding matrix for 4 class problem.

Classes	C1	C2	C3	C4	C5	C6
Class 1	1	1	1	2	2	2
Class 2	0	2	2	1	1	2
Class 3	2	0	2	0	2	1
Class 4	2	2	0	2	0	0

Table 3.5: OVR coding matrix for 4 class problem.

Classes	C1	C2	C3	C4
Class 1	1	0	0	0
Class 2	0	1	0	0
Class 3	0	0	1	0
Class 4	0	0	0	1

output indicating whether the two inputs are from the same superset or not, we believe that they are well-suited for the OVR and OVO approaches to solve a multi-class EEG classification problem. It is also worth mentioning that the classifier Λ^j is in fact a Siamese network ($G_W^j(.)$), which is trained over the two supersets $\boldsymbol{S}_0^j$ and $\boldsymbol{S}_1^j$.

3.3.3 Proposed Siamese Network for EEG Classification

As stated previously, the main idea of the proposed architecture is to decompose the multi-class classification problem into a number of binary classification problems and then employ a Siamese architecture to design each binary classifier. In this work, the OVR and OVO techniques are employed for the decomposition task and a Siamese architecture is constructed based on a CNN, which consists of two convolutional layers followed by two fully-connected layers. The network's architecture along with its hyper-parameters are shown in Figure 3.6. It should be noted that the hyper-parameters of the network are tuned through a rigorous parameter search with respect to classification accuracy over the validation set. It is worth noting that to validate the model in the development phase, a $5-$fold cross-validation approach is employed.

To train each binary classifier, Λ^j, supersets are formed according to the membership rule stated in Eq. (3.28). During the training phase, as there are cases that the number of trials in $\boldsymbol{S}_0^j$ is not equal to the one for $\boldsymbol{S}_1^j$, class weights are introduced to the loss function to compensate for the imbalanced data. To form the training package for each iteration, i.e., $([\boldsymbol{Z}_1, \boldsymbol{Z}_2, Y])$, all of the possible cases to match a trial with another trial, either from the same superset or another, are collected. The label Y is defined as follows

$$Y = \begin{cases} 1 \quad if \quad \boldsymbol{Z}_1, \boldsymbol{Z}_2 \in \boldsymbol{S}_0^j \;\; or \;\; \boldsymbol{Z}_1, \boldsymbol{Z}_2 \in \boldsymbol{S}_1^j \\ 0 \quad if \quad \text{otherwise.} \end{cases} \tag{3.29}$$

In the evaluation phase, to identify the label of an unseen trial ($\boldsymbol{Z}_{test}$), the vote of each classifier (Λ^j) is collected separately and then the final label is constructed based on the collection of votes. For each classifier, all of the cases to pair the unseen trial with training trials are collected to form $\{[\boldsymbol{Z}_{test}, \boldsymbol{Z}_{train_i}]\}_{i=1}^{p}$. Please note that in the OVO approach,

Algorithm 5 SIAMESE NETWORKS FOR BCI

Input: Training EEG trials $\{\boldsymbol{X}_k^{N_{ch}\times N_s}\}_{k=1}^{N_{trials}}$
Output: Labels of unseen trials

Obtain covariance of EEG trials
Select the coding matrix $\boldsymbol{\Sigma}$
for Each column of $\boldsymbol{\Sigma}$: $\boldsymbol{c}^j$ **do**
  Split training trials to form supersets $\boldsymbol{S}_0^j$ and $\boldsymbol{S}_1^j$ according to Eq. (3.28)
  for Each trial in $\boldsymbol{S}_0^j$: $\boldsymbol{Z}_m$ **do**
    for Each trial in $\boldsymbol{S}_1^j$: $\boldsymbol{Z}_n$ **do**
      Determine Y according to Eq. (3.29)
      Form $[\boldsymbol{Z}_m, \boldsymbol{Z}_n, Y]$
    end for
  end for
  Train a Siamese network (Λ^j or G_W^j)
end for
for An unseen trial: $\boldsymbol{Z}_{test}$ **do**
  for Each column of $\boldsymbol{\Sigma}$: $\boldsymbol{c}^j$ **do**
    for Each trial in $\boldsymbol{S}_0^j$ and $\boldsymbol{S}_1^j$ **do**
      Form $[\boldsymbol{Z}_{test}, \boldsymbol{Z}_{train_i}]$
    end for
    $y_{test}^j = \text{mode}(G_W^j([\boldsymbol{Z}_{test}, \boldsymbol{Z}_{train}]))$
  end for
  $y_{test} = \underset{i}{\text{argmin}}\Delta(\boldsymbol{y}_{test}, \boldsymbol{c}^i)$
end for

the training trials of the classes, which are labeled by "$2''$", are not participated. All the collected pairs are fed to the trained Siamese network (Λ^j) and its binary output is collected to collectively form a binary vector of votes for the unseen trial. Based on the majority of votes, the final vote of each classifier for the test trial is concluded and finally, a codeword of length equal to the number of classifiers, denoted by $\boldsymbol{y}_{test}$, is obtained. Finally, the label of the test trial is computed as

$$\underset{i}{\text{argmin}}\ \Delta(\boldsymbol{y}_{test}, \boldsymbol{c}^i), \tag{3.30}$$

where Δ and $\boldsymbol{c}^i$ denote the $L-1$ norm and the i^{th} row of the coding matrix $\boldsymbol{\Sigma}$, respectively. A detailed algorithmic workflow of our proposed Siamese architecture for single-trial MI EEG classification tasks is outlined in Algorithm 5. This finalizes the workflow of our proposed Siamese architecture for single-trial MI EEG classification.

3.3.4 Simulation Results

The proposed Siamese architecture is evaluated on the BCIC IV_{2a} dataset, which is described in Section 3.2.4. To prepare the EEG signals to be processed, the first 0.5 seconds of signals after the cue onset is dismissed to minimize the effect of activity in the visual

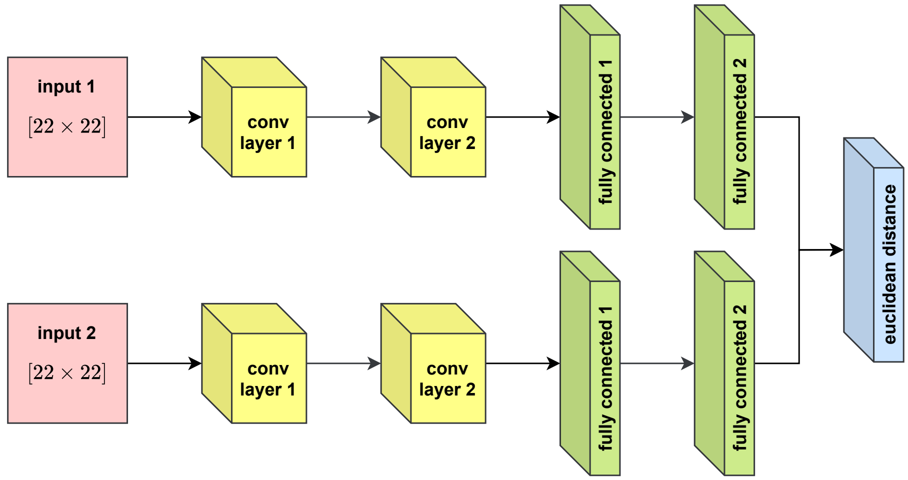


Figure 3.6: The structure of the proposed Siamese network for EEG classification. The two parallel branches, within the network, share the same parameters and hyper-parameters. Input to the network is the covariance matrix of EEG signals. **Convolutional Layer 1:** $[16 \times (3,3)$, stride=1, batch normalization, activation= Exponential Linear Unit (ELU)]; **Convolutional Layer 2:** $[32 \times (3,3)$, stride=1, batch normalization, activation= ELU]; **Fully Connected Layer 1:** [512 units, activation= Rectified Linear Unit (ReLu), dropout=0.5], and; **Fully Connected Layer 2:** [512 units, activation= ReLu].

cortex of brain on the studied motor phenomenon. Each EEG trial contains recordings from 0.5 to 2.5 seconds after the cue onset and forms a 2-dimensional matrix of size 22×500. A bandpass Butterworth filter of order 5 ($7-30$ Hz) is applied to the signals, and then, the covariance matrix of each trial is calculated as $\boldsymbol{Z}_i = (\boldsymbol{X}_i\boldsymbol{X}_i^T)/(\text{tr}(\boldsymbol{X}_i\boldsymbol{X}_i^T)$, where $\text{tr}(\cdot)$ denotes the trace of a matrix.

In the training phase, batches of 128 pairs of trials are created and fed to the proposed Siamese network. To optimize the network, Adam optimizer is employed and the learning rate is set to "$0.0001''$. Our validation results show that 25 epochs of training lead the network to an optimal balance between classification accuracy and the generalization over the studied phenomenon. It should be noted that based on our validations, the margin for similarity of two trials, m, is set to 0.5. The proposed Siamese architecture is implemented on Keras [230] (Tensorflow backend) library in Python language. In the OVR approach, as one class of data is compared against all of the other classes (3 classes), due to the imbalance of data for the "0" and "1" instances (according to Table 3.5), balancing weights of $1:3$ are introduced in the objective function to compensate for the data imbalance.

The results of our evaluations on the BCIC IV_{2a} dataset are presented in Tables 3.6 and 3.7. Table 3.6 provides a comparison between the performance of our proposed framework with renowned MI classification techniques, including FBCSP [54], SCSSP [57], and BSSFO [35, 59]. In the FBCSP, the EEG signals are decomposed into 9 different spectral bands and the frequency-specific features are obtained based on the CSP methodology. The SCSSP technique seeks to extract joint spatial and spectral features whose variance is maximized for one MI class and minimized for the other class. The BSSFO, on the other

Table 3.6: Accuracy (κ-value) of the Siamese architecture for multi-class MI classification problem.

Subject	Siamese OVR	Siamese OVO	SCSSP [57]	FBCSP [57]	FBCSP [54]	BSSFO OVO [35]	BSSFO OVA [35]
Subject 1	0.819	0.642	0.62	0.68	0.68	0.62	0.31
Subject 2	0.340	0.278	0.27	0.30	0.42	0.20	0.08
Subject 3	0.788	0.465	0.66	0.71	0.75	0.70	0.57
Subject 4	0.392	0.330	0.27	0.39	0.48	0.41	0.45
Subject 5	0.340	0.254	0.07	0.28	0.40	0.09	-0.07
Subject 6	0.389	0.351	0.26	0.25	0.27	0.15	0.08
Subject 7	0.434	0.285	0.41	0.57	0.77	0.72	0.62
Subject 8	0.705	0.611	0.59	0.59	0.76	0.58	0.30
Subject 9	0.778	0.632	0.66	0.54	0.61	0.65	0.55
Average	0.554	0.428	0.42	0.48	0.57	0.46	0.32

Table 3.7: Accuracy (%) for binary classification of 4 MI tasks.

Subject	1 vs. 2	1 vs. 3	1 vs. 4	2 vs. 3	2 vs. 4	3 vs.4
Subject 1	79.44	91.00	91.71	93.60	100.0	59.86
Subject 2	63.88	65.78	64.80	62.92	62.56	63.99
Subject 3	88.20	83.00	89.77	86.12	92.84	64.64
Subject 4	53.93	65.55	76.47	74.41	70.56	54.11
Subject 5	54.94	56.26	58.51	61.53	54.49	53.81
Subject 6	65.67	61.56	65.64	65.69	64.61	62.43
Subject 7	51.69	78.46	78.29	74.83	73.37	56.57
Subject 8	91.81	78.58	93.15	76.89	81.25	73.86
Subject 9	82.76	94.30	97.24	76.06	85.75	84.31
Average	70.25	74.94	79.5	74.67	76.16	63.74

hand, employs a Bayesian optimization framework to derive subject-specific spectral filters to extract the most informative CSP-based features. In addition, the results in Table 3.7 reflect the efficacy of our proposed deep learning-based Siamese architecture for binary classification of EEG signals. It should be noted that the results in Table 3.7 represent the classification accuracy, whereas the results in Table 3.6 show Kappa coefficient (κ) for classification accuracy. Kappa coefficient reveals the performance of a classifier compared to random labeling of unseen trials and is calculated as

$$\kappa = (P_s - P_r)/(1 - P_r), \tag{3.31}$$

where P_s is the probability of correct classification for the system and P_r is the probability of random labeling of the unseen trials. It is worth adding that reporting the classification accuracy in κ coefficient is proposed by the BCI competition, and thus the same procedure is followed to facilitate the comparison of different works. Moreover, the κ coefficient is commonly employed to report the accuracy in multi-class problems, therefore, the results in Table 3.6 are in κ format and the ones in Table 3.7 only reflect the portion of correct predictions. One of the key insights reflected by Table 3.6 is that the OVR approach outperforms the OVO one, which is due to the fact that in OVR, each class is compared with the rest of other classes and the network learns how to collectively distinguish one class from other classes. It should be highlighted that in this work the proposed Siamese architecture is evaluated in its most basic format to provide a proof-of-concept for this idea, thus several modifications and enhancements could be applied to the proposed technique to outperform the state-of-the-art results in this field, which will be the basis of our future works.

3.4 Optimal Dimensionality Reduction for EEG Signals

As discussed earlier, a typical signal processing pipeline for EEG signals consists of tow major blocks, referred to as spectral filtering and spatial filtering, and majority of research works in this domain investigate different methodologies, and architectures for either of theses two blocks to enhance the clarity of the signals and maximize the amount of exploitable information from them.

- **Spatial Filtering**: In this regard, several methods are proposed in the literature among which the CSP and Riemannian manifold learning are known to be among the most effective techniques in sensorimotor signals analysis. CSP derives spatial filters, which map the recordings into a new coordinate where a higher weight is given to the channels that demonstrate the ERD and ERS waveforms. The superior performance of CSP in analyzing motor imagery signals and achieving high classification accuracy has motivated several numbers of research works to further extend this technique to achieve even higher accuracies. A number of these extensions include FBCSP [54], and Regularized Common Spatial Patterns (RCSP) [55]. On the other hand, as the EEG recordings are real valued and their covariance matrices are Symmetric Positive-definite (SPD), the space of SPD matrices with Riemannian distance define a Riemannian manifold which recently has received substantial attention for high dimensional EEG classification [231]. Recent works on EEG classification based on Riemannian manifold learning [232, 233] have shown superior results.

- **Spectral Filtering**: It is shown that decomposition of the EEG signals into a few constructing spectral components results in the enhancement of the overall accuracy of the classification system. Among different approaches in decomposing the signals, employment of spectral filters and Wavelet decomposition are the most regarded ones. Works such as [59] have used the spectral filters and filterbanks to decompose the signals into a number of spectral components. These works are based on the fact that different types of mental tasks occur in different frequency ranges and thus the decomposition helps to focus on the intended mental task. On the other hand, the Wavelet decomposition methods provide a temporal-spectral means to decompose the EEG recordings into a few components [234, 235]. In this approach, each component is still bounded in a specific frequency band, however, the bandwidth of each component is a function of the sampling frequency of the signal as well as the levels of decomposition. Although the Wavelet decomposition methods normally benefit from lower latency and less distortion imposed on the signal, they provide limited flexibility to control the bandwidth of each components to focus on a certain spectral range.

Although several works have investigated the great potentials of combining the Wavelet decomposition methods with the CSP spatial filtering and feature extraction method, to best of our knowledge, there is no research on coupling the Wavelet methods with the Riemannian learning methods. The paper addresses this gap. Several test scenarios were conducted to evaluate the proposed Wavelet Riemannian spectral-spatial filtering (WvRiem) and compare its performance with filterbank-CSPs, and Wavelet-CSPs.

3.4.1 Problem Formulation

The problem of single trial EEG classification is approached based on the available EEG dataset, which is denoted by $\{\boldsymbol{X}_i \in \mathbb{R}^{N_{\text{ch}} \times N_{\text{t}}}\}$, for $(1 \leq i \leq N_{\text{Trial}})$, where N_{Trial} is the total number of training trials; N_{ch} is the number of EEG channels (electrodes), and; N_{t} is the number of time samples collected from each electrode in one trial. The label of each trial is denoted by Ω which could be "right hand MI" or "left hand MI" and the whole training dataset is represented by $\{(\boldsymbol{X}_i, \Omega_i)\}$, for $(1 \leq i \leq N_{\text{Trial}})$. The employed signals are initially preprocessed by removing the high-frequency interferences by band-pass filtering and also removing the effect of power-line on the recordings by notch filtering. This completes a brief presentation of the problem at hand. Next, the WvRiem is discussed in details.

Riemannian Geometry

The spatial covariance of one trial of EEG recordings is denoted by $\boldsymbol{\Sigma}$. Since the EEG recordings are real-valued and their covariance matrix turns out to be SPD, and the fact that SPD matrices equipped with Riemannian distance are a differentiable Riemannian manifold, the Riemannian geometry could be employed to analyze the SPD matrices. To this end, the space of symmetric matrices is defined as $\mathscr{S}(N_{\text{ch}}) = \{\boldsymbol{\Sigma} \in \mathbb{R}^{N_{\text{ch}} \times N_{\text{ch}}}, \boldsymbol{\Sigma} = \boldsymbol{\Sigma}^T\}$, and the space of positive-definite matrices is defined as $\mathscr{C}(N_{\text{ch}}) = \{\boldsymbol{\Sigma} \in \mathbb{R}^{N_{\text{ch}} \in N_{\text{ch}}}, \mathbf{u}^T \boldsymbol{\Sigma} \mathbf{u} > 0, \forall \mathbf{u} \in \mathbb{R}^{N}_{\text{ch}}\}$. Thus, the space of SPD matrices is defined as $\mathscr{SPD}(N_{\text{ch}}) = \mathscr{S}(N_{\text{ch}}) \cap \mathscr{C}(N_{\text{ch}})$. As the SPD matrix could fit a differentiable Riemannian manifold, most of the mathematical concepts from the Riemannian geometry domain could be borrowed and applied on $\mathscr{SPD}(N_{\text{ch}})$.

The Riemannian distance between two matrices $\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2 \in \mathscr{SPD}(N_{\text{ch}})$ is defined as follows

$$\delta_R(\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2) = ||\log(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_2)||_F \;=\; \left[\sum_{i=1}^{N_{\text{ch}}} \log^2 \beta_i\right]^{1/2}, \tag{3.32}$$

where $||.||_F$ is the Frobenius norm of a matrix and β_i is the i-th real eigenvalue of $\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_2$. One of the most important features of the Riemannian distance is its invariance to bilinear transformation

$$\delta_R(\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2) = \delta_R(\boldsymbol{W}^T\boldsymbol{\Sigma}_1\boldsymbol{W}, \boldsymbol{W}^T\boldsymbol{\Sigma}_2\boldsymbol{W}), \tag{3.33}$$

where the transformation matrix $\boldsymbol{W} \in \mathbb{R}^{N_{\text{ch}} \times N_{\text{ch}}}$ is invertible. The mean of SPD matrices plays an important role in classification and is defined as the point $\boldsymbol{\Sigma}_R \in \mathscr{SPD}(N_{\text{ch}})$, which has a minimum sum of the squared distances to all SPD matrices in dataset $\mathscr{C}$. The

Algorithm 6 LEVEL-BASED EEG CLASSIFICATION VIA WCSP

Input: EEG recordings $\{\boldsymbol{X}_i\}_{i=1}^{N_{\text{Trial}}}$ and their corresponding labels $\{\Omega_i\}_{i=1}^{N_{\text{Trial}}}$
Output: The labels of the unseen data

Decompose the signals $\{\boldsymbol{X}_i; 1 \leq i \leq N_{\text{Trial}}\}$ into l level of details.
for $l \in$\{The number of decomposition levels\} **do**
    ——— In Training Phase ———
    Form a number of binary classification problems based on the preferred coding matrix.
    Apply CSP spatial filtering on the two formed super-sets.
    Reduce the number of channels in data to $2m$.
    Extract CSP features from the transformed data.
    Train a classifier based on the training features.
end for
——— In Validation Phase ———
Obtain the best classification accuracy for different levels via k-fold cross validation and select the best level (l^*).
——— In Testing Phase ———
Obtain the wavelet transform of the unseen trials.
Extract the details of signals in l^* level.
Apply the spatial filters for the l^* level on the signals.
Extract corresponding features and perform classification.

Riemannian mean is

$$\boldsymbol{R}_R = \arg \min_{\boldsymbol{\Sigma} \in \mathscr{SPD}(N_{\text{ch}})} \sum_{\boldsymbol{\Sigma}_i \in \mathscr{C}(N_{\text{ch}})} \delta_R^2(\boldsymbol{\Sigma}, \boldsymbol{\Sigma}_i). \tag{3.34}$$

3.4.2 Classification Scheme

As the problem in hand constitutes of multiple number of classes, and on the other hand, both the CSP and Riemannian methods are suitable for binary problems, the OVR, ECOC, and TECOC classification schemes are employed.

Wavelet-CSP Methodology

In this methodology, the overall idea is to decompose the recordings into several levels of detail components and then shrink the size of the features space by identifying the most informative level in the validation phase. Then, the classification of unseen trials is based on the level-specific features. This process is performed on the training data and the best result (i.e., the best level of decomposing which yields the best accuracy) is selected. K-fold cross validation is employed to reveal the most informative level of decomposition, as one of the hyper-parameters for the testing stage. Algorithm 6 summarizes the WCSP methodology.

Algorithm 7 LEVEL-BASED EEG CLASSIFICATION VIA WVRIEM

Input: EEG recordings $\{\boldsymbol{X}_i\}_{i=1}^{N_{\text{Trial}}}$ and their corresponding labels $\{\Omega_i\}_{i=1}^{N_{\text{Trial}}}$
Output: The labels of the unseen data

Decompose the signals $\{\boldsymbol{X}_i; 1 \leq i \leq N_{\text{Trial}}\}$ into l level of details.
——— In Training Phase ———
for $l \in$\{The number of decomposition levels\} **do**
    Form a number of binary classification problems based on the preferred coding matrix.
    Obtain the Riemannian mean of the two super-sets (R_1, R_2).
    Obtain the bilinear transformation $(\boldsymbol{W})$ as in $[\boldsymbol{W}, \mathbf{D}] = eig(R_1, R_1 + R_2)$.
    Reduce the number of channels in data to $2m$ based on $\mathbf{D}$.
    Apply $\boldsymbol{W}$ on the spatial covariance matrix of all trials.
    Calculate the set-mean of the transformed covariances (R_{s1}^{l}, R_{s2}^{l}).
end for
——— In Validation Phase ———
Obtain the best classification accuracy for different levels and select the best one (l^*).
——— In Testing Phase ———
Obtain the wavelet transform of the unseen trials.
Extract the details of signals in l^* level and calculate its covariance matrix $(\boldsymbol{\Sigma}_{l^*})$.
Apply the bilinear transformation $(\boldsymbol{W})$ for the l^* level on the covariance matrix.
Compare the Riemannian distances; $\delta_R(W^*\boldsymbol{\Sigma}_{l^*}W^{*'}, R_{s1}^{l^*})$ and $\delta_R(W^*\boldsymbol{\Sigma}_{l^*}W^{*'}, R_{s2}^{l^*})$.

Wavelet-Riemannian (WvRiem) Methodology

Similar to the Wavelet-CSP methodology, in this scheme also, the goal is to decompose the signals into a number of components and then employ the Riemannian methods to perform the classification for the unlabeled trials. In this case also, the k-fold cross validation is performed on the training data to identify the best level (l^*) for classification for each individual. It is worth mentioning that the classification for Riemannian method is based on the distance between the unlabeled trial and the Riemannian mean of the two super-sets and thus, no specific classifier is employed in this part. The WvRiem method is summarized in Algorithm 7.

3.4.3 Simulation Results

To evaluate the performance of the proposed processing pipelines, the dataset BCIC-IV_{2a}, as described in Section 3.2.4 is employed. As it is presented in Algorithms 6 and 7, the first step is to identify the coding matrix for the classification part. Since the "Naive Bayes Classifiers" is employed for the binary classification step, which has shown superior performance in [54], for each trial, the classifier outputs two probabilities associated with each class. By comparing the probabilities of the classes labeled with "1" and selecting the maximum one, the final label of the unseen trial is obtained.

To measure the performance of classification and as it is suggested by the BCI Competition, the kappa (κ) score is employed. Kappa score is calculated according to $\kappa =$

Table 3.8: Classification accuracies in Kappa score on BCIC-IV_{2a}.

Subject	[57]	FBCSP OVR	FBCSP TECOC	WCSP OVR	WCSP TECOC	WvRiem OVR	WvRiem TECOC
1	0.501	0.398	0.403	0.648	0.648	0.477	0.694
2	0.210	0.199	0.153	0.296	0.287	0.245	0.278
3	0.621	0.551	0.597	0.648	0.634	0.644	0.662
4	0.314	0.375	0.440	0.468	0.500	0.380	0.514
5	0.000	0.227	0.204	0.222	0.231	0.213	0.250
6	0.121	0.134	0.153	0.264	0.301	0.250	0.236
7	0.334	0.398	0.426	0.528	0.560	0.472	0.639
8	0.533	0.634	0.699	0.713	0.676	0.560	0.722
9	0.633	0.574	0.569	0.597	0.676	0.639	0.653
Average	0.363	0.433	0.407	0.487	0.501	0.431	0.516

$\frac{CCR-P_{rand}}{1-P_{rand}}$, where CCR stands for "Correct Classification Rate" and P_{rand} is the probability of random selection which in dataset is equal to 0.25. Kappa score provides a measure on how better than random selection the system has performed. To implement the wavelet decomposition step, several wavelet functions were applied which among them "daubechies (db)5", "db9", "db11", and "haar" yielded the best results. To report the classification accuracy for each subject, the maximum performance for different wavelet functions is deployed. Moreover, the results of the cross-validation procedure revealed that 5 levels of decomposition leads to the best classification accuracies.

The performances of the proposed Wavelet-CSP and Wavelet-Riemannian pipelines are calculated in Kappa score and are presented in Table 3.8. In order to compare the results with a baseline, the filterbank-CSP method is employed. In this method, a filterbank of 9 spectral filters with frequency ranges of $4-8, 8-12, \ldots, 36-40$ Hz is applied on the signals to decompose the recordings into 9 components. Then, the CSP filters for each component are derived and 9 different classifiers are trained. The best classification performance for unseen data across all the 9 classifiers is reported in this table. To further investigate the effect of the level-based classification procedure and to understand its benefits in terms of classification accuracy and computational cost, the following testing scenarios are conducted. First, the performance of the network is examined across two cases: (i) All decomposition levels are employed to train the classifier, and; (ii) Only the best (identified through cross-validation step) level of decomposition (l^*) are employed. The results are presented in Table 3.9. As it is observed, the reduced dimension of the features (Case (ii)), which is based on employing the best decomposition level, yields better results than Case (i). It is worth mentioning that the improved accuracy comes with the benefit of lower computational costs. Within the context of Riemannian manifold learning, classification is performed by comparing the Riemannian distance between the covariance matrix of unlabeled trial with the mean covariance matrices of the two super sets. For Case (i), i.e., when all the levels of decomposition are participated in the classification, the mean distances across all the levels is compared to derive the final class label.

An important point, which is common in all the presented experimental results, is that the proposed WvRiem method is only evaluated in the multi-class classification problems. In multi-class problems, the classification step acts as a bottleneck in the system as it can

Table 3.9: Performance comparison between participation of all or the best component in the classification step.

Subjects	WCSP + OVR		WCSP + TECOC		WvRiem + OVR		WvRiem + TECOC	
	All	Best	All	Best	All	Best	All	Best
1	0.625	0.648	0.630	0.648	0.431	0.477	0.690	0.694
2	0.264	0.296	0.277	0.287	0.245	0.245	0.278	0.278
3	0.643	0.648	0.611	0.634	0.588	0.644	0.616	0.662
4	0.449	0.468	0.462	0.500	0.333	0.380	0.500	0.514
5	0.222	0.222	0.152	0.231	0.171	0.213	0.046	0.250
6	0.213	0.264	0.259	0.301	0.194	0.250	0.250	0.236
7	0.551	0.528	0.527	0.560	0.417	0.472	0.495	0.639
8	0.713	0.713	0.685	0.676	0.491	0.560	0.615	0.722
9	0.611	0.597	0.583	0.676	0.625	0.639	0.625	0.653
Average	0.477	0.487	0.465	0.501	0.388	0.431	0.457	0.516

severely affect the overall performance of the system. Thus, the authors find it necessary to evaluate the proposed method over binary classification problems. In this regard, 6 binary classification problems are defined over the same previous dataset and the framework was assessed over 9 subjects. The results of this experiment are presented in Table 3.10. As it can be observed, the WvRiem method also outperforms its CSP-based counterpart in binary classification problems. As it is understood from the results, the combination of Wavelet decomposition methods and Riemannian manifold learning (WvRiem) outperforms the combination of Wavelet decomposition methods with CSP filtering technique (WCSP). For the multi-class problems also, WvRiem coupled with Ternary-ECOC classifiers yields better classification accuracy compared to its counterparts. It is worth mentioning that the employed Riemannian learning method is the most basic one and the classification is only based on distance and no feature generation and classification is applied. Thus, this is a great motivation to investigate this area more further.

3.5 Summary

In this chapter, a series of contributions in data-driven methods for EEG processing and BCI-systems were covered. In particular, an optimization technique based on Bayesian methods was introduced that derives subject-specific spatial and spectral filters for EEG-based BCI systems. Thereafter, a classification scheme referred to as the TECOC was introduced, which enhances the classification accuracy of BCI systems in multiclass classification problems. Then, to leverage the massive learning capacity of neural networks, and to overcome the issue with training deep neural networks with small medical datasets, a novel neural architecture, referred to as Siamese neural networks, was employed for the first time in the BCI domain. Finally, a dimensionality reduction technique for EEG signals based on wavelet transformations and Riemannian methods was introduced, which can play an important role in developing real-time EEG processing frameworks. In the next chapter,

Table 3.10: Performance comparison of the two WCSP and WvRiem methods over the binary classification problems

WCSP WvRiem	1 vs. 2	1 vs. 3	1 vs. 4	2 vs. 3	2 vs. 4	3 vs. 4
Subject 1	0.792 0.792	0.916 0.945	0.903 0.958	0.945 0.972	1.000 1.000	0.445 0.361
Subject 2	0.126 0.139	0.445 0.612	0.237 0.264	0.403 0.597	0.319 0.264	0.514 0.583
Subject 3	0.916 0.889	0.916 0.889	0.903 0.834	0.792 0.876	0.903 0.972	0.334 0.403
Subject 4	0.375 0.445	0.805 0.681	0.639 0.750	0.750 0.820	0.583 0.639	0.501 0.486
Subject 5	0.375 0.348	0.208 0.501	0.390 0.430	0.097 0.348	0.348 0.486	0.361 0.055
Subject 6	0.319 0.292	0.237 0.541	0.306 0.348	0.25 0.417	0.417 0.306	0.528 0.514
Subject 7	0.723 0.403	0.987 0.805	0.861 0.847	1.000 0.805	0.903 0.889	0.501 0.597
Subject 8	0.916 0.903	0.834 0.694	0.916 0.945	0.736 0.723	0.805 0.765	0.667 0.612
Subject 9	0.849 0.889	0.916 0.889	0.931 0.972	0.654 0.430	0.667 0.667	0.754 0.792

the novelties of the thesis in developing data-driven frameworks for hand tremor management, with applications in ADs, robotic rehabilitation technologies, and robotic-assisted surgeries will be covered. To be more specific, next chapter will revolve around the developed computational techniques to estimate different underlying components of hand motion in the patients affected with age-related neurological disorders.

Chapter 4

Data-driven Methods for Pathological Hand Tremor Estimation

4.1 Introduction

In Chapter 3, the contributions of the thesis in data-driven signal processing and machine learning frameworks for EEG-based BCI technologies were presented. Although the discussed contributions are applicable independently in various BCI use cases, they can also be integrated into an innovative design for AD systems, e.g., when BCI serves as a building block in the FES system as discussed in Section 1.1.3. The focus of this chapter, on the other hand, is on the contributions of the thesis towards innovative and efficient data-driven models for PHT processing. The proposed methodologies in this chapter are of great importance for developing more accurate and efficient ADs, with application in robotic rehabilitation systems, tele-robotic surgery technologies, smart spoons, and the FES system as described in Section 1.1.3. The rest of this chapter is organized as follows: First, a multi-rate and adaptive data-driven model based on wavelet transformations and Kalman filtering is proposed to analyze the PHT kinematics signals and distinguish the voluntary and involuntary components of hand motion; Second, a DL-based solution is introduced, which stands up to address the challenge with the unavailability of a ground truth signal for voluntary actions of the individuals with PHT, and; Finally, the chapter is concluded by examining the proposed DL-based approach over a number of smaller PHT datasets to verify the efficacy of the proposed model over various datasets.

4.2 The WAKE Framework

In this section, a processing framework is proposed, which aims at employing the residual functionalities of the patients' limbs to control an AD. To be more specific, the proposed framework in this section is capable of analyzing hand motion signals of the patients and estimate/extract the voluntary motion out of the noise (tremor) contaminated signals. Accurate separation of the two components is of great importance in rehabilitation devices and more specifically, plays an important role in the overall performance of the hBCI system. The need for a new on-line and real-time tremor estimation/extraction framework is

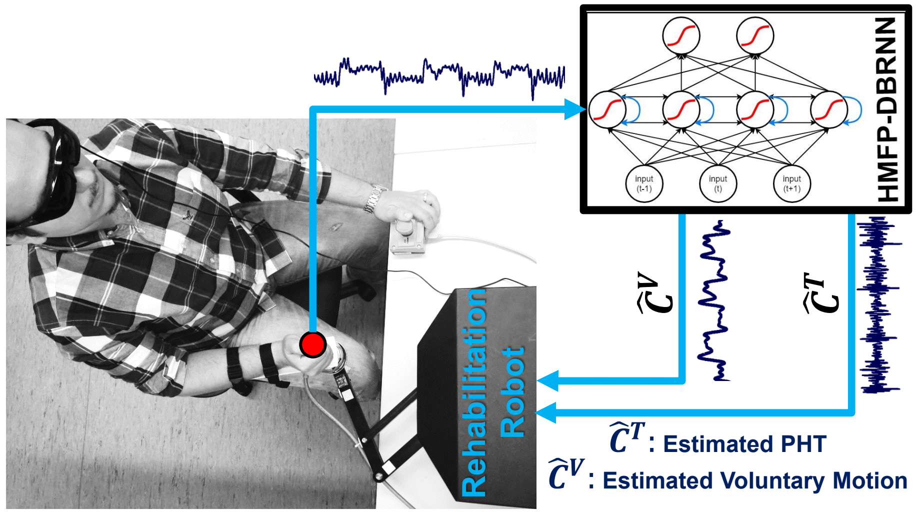


Figure 4.1: Block-diagram of an Augmented Haptic Rehabilitation (AHR) system, where tremor extraction is required to develop a safe haptics-enabled robotic rehabilitation system.

addressed by incorporating spectra-temporal movement information and propose a novel framework referred to as Wavelet Adaptive Kalman Tremor Extraction (WAKE). The proposed framework decomposes the hand motion signal into several spectra-temporal components via wavelet transforms and then provides myopic predictions on the tremor by means of an adaptive auto-adjustable Kalman filtering framework. The WAKE framework incorporates a multi-rate scheme to extract the tremor out of the measurement signal by utilizing KF and wavelet transforms in an adaptive, iterative, and optimized fashion. The proposed WAKE framework consists of the following two schemes running in parallel but with different rates: (a) *Real-time Tremor Prediction (RTP)*, which operates in an on-line manner and provides predictions for the tremor signal in the next time instance, and; (b) *Hyper Parameter Adjustment (HPA)*, which operates in slower rate and optimizes hyper-parameters of the filter to boost the performance and accuracy of the RTP scheme.

In summary, the proposed WAKE framework runs in real-time and provides myopic (one-step ahead) predictions of the pathological tremor signal via an adaptive and multi-rate mechanism making it suitable for implementation on hardware and real world applications. In the next section, the problem in hand is formulated.

4.2.1 Problem Formulation

In this section, the mathematical background required for development of the proposed WAKE framework are briefly provided. Throughout the thesis, the hand movement of a patient with pathological tremor is modeled as

$$m(n) = m_v(n) + m_i(n), \tag{4.1}$$

where $m(n)$ represents the composite motion signal (measurement); $m_v(n)$ and $m_i(n)$ are the voluntary and involuntary (tremor) components of the motion, respectively; and n denotes the time index. In the proposed WAKE framework, wavelet transforms are employed to obtain multi-scale decomposition of the measurement $m(n)$. This is similar in nature to the Fourier transform used in the FLC-based tremor extraction methodologies. However, unlike sinusoidal functions, a wavelet and the ones generated from it are localized in space, therefore, providing a mechanism to approximate $m(n)$ by a series of scaled and translated versions of these localized functions. In other words, The wavelet transform decomposes the signal into different scales/resolutions. Lower scales provide more details of high frequency components while higher scales provide overall features associated with low frequency components. Before presenting the proposed WAKE framework, the wavelet decomposition technique, i.e., the spectra-temporal analysis, is briefly discussed in the next section.

Spectra-temporal Signal Analysis

Various spectra-temporal signal analysis methods have been investigated in literature including the STFT, WVT, CWD, and CWT [236, 237], among which the latter is more favorable due to providing high resolution for high-frequency signals. A wavelet transform represents a signal into different scales and dilations of a "finite-length" and "fast-decaying" oscillating waveform known as the wavelet function (mother wavelet) and scaling function (father wavelet). Mathematically speaking, a mother wavelet $\psi(t)$ should be a square integrable function, and satisfy the admissibility condition and the regularity condition, which requires $\psi(t)$ to be fast decaying or be non-zero only on a finite interval. To form the wavelet transform of a given signal, different dilations and scales of the mother wavelet function $\psi_{j,u}(t)$ are applied to the signal as

$$\psi_{j,u}(t) = s_0^{-j/2}\psi(s_0^{-j}t - u\tau_0), \quad j,u \in \mathbb{Z}, \tag{4.2}$$

where u and j represent dilation, and scale of the mother wavelet function, respectively. Terms $s_0 > 1$ and τ_0 are fixed dilation and translation factors [237]. The DWT is then defined by

$$\boldsymbol{T}_x(j,u) = \int_{-\infty}^{\infty} \boldsymbol{x}(t)\psi^*_{j,u}(t)dt. \tag{4.3}$$

If the family of wavelets $\psi_{j,u}(t)$ form an orthogonal basis, the signal $\boldsymbol{x}(t)$ could be recovered from its discrete wavelet decomposition ($\boldsymbol{T}_x(j,u)$) as in

$$\boldsymbol{x}(t) = \frac{1}{c_\psi}\sum_{j,u\in Z}\boldsymbol{T}_x(j,u)\psi_{j,u}(t). \tag{4.4}$$

The father (scaling) function, which has also the same shape as of the mother wavelet, represents smoothed image of the signal and is defined as follows

$$\phi_{j,u}(t) = s_0^{-j/2}\phi(s_0^{-j}t - u\tau_0), \quad j,u \in \mathbb{Z}, \tag{4.5}$$

where $\int_{-\infty}^{\infty}\phi_{0,0}(t) = 1$, and $\phi_{0,0} = \phi$. The mother wavelet acts as a high-pass filter, so its coefficients ($\boldsymbol{T}_x(j,u)$) represent the *details* of the signal, while the father wavelet behaves

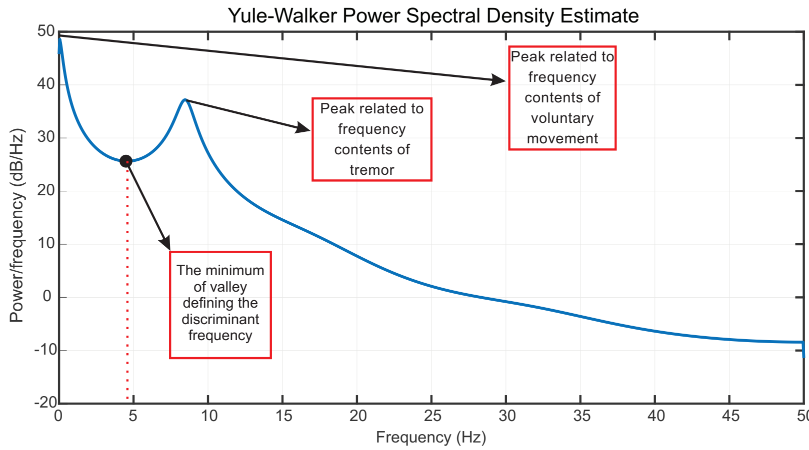


Figure 4.2: The power spectrum estimation for a sample measurement signal containing voluntary movement and action tremor.

as a low-pass filter which provides the *approximations* of the signal. To derive the approximation coefficients, $\boldsymbol{S}_x(j,u)$, the father wavelet function is convolved with the signal as $\boldsymbol{S}_x(j,u) = \int_{-\infty}^{\infty} \boldsymbol{x}(t)\phi_{j,u}(t)dt$. The approximation coefficients at a specific scale j represent the discrete approximation of the signal at that scale. A continuous approximation of the signal at scale j is derived by summing a sequence of father wavelets at this scale factored by the approximation coefficients as follows $\hat{\boldsymbol{x}}_j(t) = \sum_{u=-\infty}^{\infty} \boldsymbol{S}_x(j,u)\phi_{j,u}(t)$, where $\hat{\boldsymbol{x}}_j(t)$ is a smooth, scaling-function-dependent version of the signal $\boldsymbol{x}(t)$ at scale index j.

4.2.2 Methodology

In this section, the proposed framework, which is designed to extract the tremor signal from the raw measurements consisting of both voluntary and involuntary components of motions is presented. To accurately predict the tremor in the next time instant and in real-time, WAKE introduces a framework consisting of the following two multi-rate filtering schemes operating in parallel and provide a self-adjustable tremor extraction engine:

- **Real-time Tremor Prediction (RTP) Scheme:** which processes the raw sensory data in real-time and predicts the value of the voluntary motion in the next time instant.

- **Hyper-Parameter Adjustment (HPA) Scheme:** which operates in slower rate than the RTP scheme and performs post-processing on the previous time samples to derive optimal values for the hyper-parameters. The optimized parameters provided by the HPA scheme result in higher performance in tremor extraction and prediction.

In the following subsections, the details of the HPA and the RTP schemes are discussed respectively.

Hyper-parameter Adjustment (HPA) Scheme

In the proposed multi-rate WAKE framework, the HPA scheme has slower rate of execution in comparison to the RTP as it performs post-processing on the previous data samples and extracts a number of hyper-parameters which are fed into the RTP scheme. In particular, the HPA scheme performs the analysis over a pre-defined window of length L. At each time index n, the last L measured samples are combined in a vector denoted by $\boldsymbol{m}_L(n) \triangleq [m(n-L+1),\ldots,m(n)]^T$, where superscript T denotes transpose operator. Please note that Term $\boldsymbol{m}_L(n)$ in Eq. (5.11) is a vector of $(L\times 1)$ dimension, as it is constructed by stacking the last L measured samples ($m(n-L+1)$ to $m(n)$). The power spectrum estimation of the windowed signal $\boldsymbol{m}_L(n)$ is then computed based on the Yul-Walker approach as follows

$$\boldsymbol{p}_L(f) = \frac{\hat{\sigma}_w^2}{|1+\sum_{u=1}^{p}\hat{a}_u e^{-j2\pi f u}|^2}. \tag{4.6}$$

The Yule-Walker method is grounded on the hypothesize that the signal, which its power spectrum is desired (i.e., $\boldsymbol{m}_L(n)$), is the output of a linear time-invariant (LTI) system where the input to the system is a zero-mean white noise. In this case, $\hat{a}_u$, for $(1\leq u\leq p)$, is the estimated Autoregressive (AR) parameter of the output signal using the Levinson Durbin algorithm, and $\hat{\sigma}_w^2$ is the p^{th}-order Minimum Mean-Square Estimate (MMSE) of the variance of the zero-mean white noise provided as the input to the system which is computed as $\hat{\sigma}_w^2 = \gamma_L(0)\prod_{\tilde{a}u=1}^{p}[1-|\hat{a}_{\tilde{a}u}|^2]$, where the biased autocorrelation estimate $\gamma_L(0)$ is given by $\gamma_L(0) = \frac{1}{L}\boldsymbol{m}_L^H(n)\boldsymbol{m}_L(n)$. Superscript H denotes hermitian transpose operator. Figure 4.2 illustrates one sample of the power spectrum of a hand motion signal which consists of both tremor and voluntary motions. As can be seen from Figure 4.2 and as shown in relevant literature (such as in [81, 126, 238]), the frequency band of the tremor is completely distinct from that of the voluntary movement. This distinction results in a valley (between the frequency contents of tremor and that of the voluntary movements) in the power spectrum of the signal. In this work, the central frequency of the valley that separates the frequency contents of the tremor from that of the voluntary motion is called as the *cut-off frequency* (f_d).

The cut-off frequency f_d can be used to form and extract a ground truth for the tremor signal denoted by $\boldsymbol{m}_{i_L}^{(\text{GT})}(n)$. For this purpose, typically [81, 126, 238], a sharp high-pass filtering is applied on the windowed version $\boldsymbol{m}_L(n)$ of the signal with cut-off frequency f_d. The sharp high-pass filtering approach is described in Algorithm 8. However, in the existing methodologies a pre-defined and fixed value of f_d is used to extract the ground truth. In contrary, the WAKE proposes to incorporate an autonomous tremor frequency band detection algorithm by capitalizing on the smoothness of the curve of the power spectrum density, and the apparent distinction between the frequency contents of tremor versus that of the voluntary motion. The proposed autonomous detection algorithm computes the minimum value of the valley in the power spectrum density curve which is a representative of the cut-off frequency f_d. Thus, the cut-off frequency f_d is not a pre-defined or fixed value, instead, its estimated value is updated in a real-time fashion at each iteration of the HPA scheme.

After high-pass filtering the measurement signal, and extracting the ground truth for the

Algorithm 8 SHARP HIGH-PASS FILTERING METHODOLOGY ACCORDING TO [81]

Input: The measurement signal: $\boldsymbol{m}_L(n)$.

Output: Hyper-Parameters: The ground truth for the tremor signal.

1: Calculate the Fast Fourier Transform (FFT), denoted by $\boldsymbol{F}(\boldsymbol{m}_L(n))$, of the input $\boldsymbol{m}_L(n)$.

2: Construct an indexing vector with the same length as of $\boldsymbol{F}(\boldsymbol{m}_L(n))$, i.e., the index vector has ones for frequencies greater than f_d and zeros for the rest.

3: Multiply the indexing matrix by $\boldsymbol{F}(\boldsymbol{m}_L(n))$ in an element-wise fashion; $\Big[\boldsymbol{F}(\boldsymbol{m}_L(n))\Big]_{indexed}$.

4: Apply inverse-FFT to $\Big[\boldsymbol{F}(\boldsymbol{m}_L(n))\Big]_{indexed}$.

tremor signal $\boldsymbol{m}_{i_L}^{(\mathrm{GT})}(n)$, the ground truth for voluntary movement is derived as $\boldsymbol{m}_{v_L}^{(\mathrm{GT})}(n) = \boldsymbol{m}_L(n) - \boldsymbol{m}_{i_L}^{(\mathrm{GT})}(n)$. The final step of the HPA scheme is to find the optimum values for the adaptive weights denoted by $\boldsymbol{w}_L(n)$ (i.e., Hyper-parameters) for the current iteration which is then used by the RTP scheme to fuse wavelet approximations and construct the voluntary movement. In other words, the $\boldsymbol{w}_L(n)$ vector defines a weight for each approximation in a way that the weighted mixture of the approximations results in an optimized reconstructed voluntary signal. To achieve this goal, a wavelet decomposition is applied to the windowed signal $\boldsymbol{m}_L(n)$ for extracting J number of approximations. The windowed signal $\boldsymbol{m}_L(n)$ can then be represented by a combined series expansion using both the approximation coefficients and the wavelet (detail) coefficients as follows

$$\boldsymbol{m}_L(n) = \sum_{u=-\infty}^{\infty} \boldsymbol{S}(J,u)\phi_{J,u}(n) + \sum_{j=1}^{J}\sum_{u=-\infty}^{\infty} \boldsymbol{T}(j,u)\psi_{j,u}(n). \tag{4.7}$$

In order to reconstruct the j^{th} approximation of the signal $\boldsymbol{a}_j$, for $(1 \leq j \leq J)$, according to the multilevel behavior of the wavelet transforms, the detail and approximation coefficients of the signal from the $(j+1)^{\text{th}}$ level of decomposition should be taken into account. On the other hand, as the DWT requires having a discrete time signal as an input which has fixed sampling rate, the approximation coefficients are extracted as in $\boldsymbol{S}(j,u) = \sum_{n=1}^{L} \boldsymbol{m}_L(n)\phi_{j,u}(n)$, and the detail coefficients as in $\boldsymbol{T}(j,u) = \sum_{n=1}^{L} \boldsymbol{m}_L(n)\psi_{j,u}(n)$. The approximation signal in level j (i.e., $\boldsymbol{a}_j$), for $(1 \leq j \leq J)$, can be reconstructed as follows,

$$\boldsymbol{a}_j(n) = \sum_{u=-\infty}^{\infty} \boldsymbol{S}(j+1,u)\phi_{j+1,u}(n) + \sum_{u=1}^{\infty} \boldsymbol{T}(j+1,u)\psi_{j+1,u}(n). \tag{4.8}$$

Reconstructed approximations at all levels are represented in compact form as matrix $\boldsymbol{A}_L(n) = \{\boldsymbol{a}_{(j)}\}_{1 \leq j \leq J} \in \mathbb{R}^{L \times J}$.

Please note that extracted approximation coefficients together with the detailed coefficients form the original signal (Eq. (4.7)), while the reconstructed approximations provide a smoothed version of the signal without the need for the detail coefficients. The levels of

Algorithm 9 THE HPA FRAMEWORK

Input: The measurement signal: $\boldsymbol{m}_L(n)$.

Output: Hyper-Parameters: Optimized weight vector $\boldsymbol{w}_L(n)$; Discriminant frequency f_d; and Estimated ground truth vector $\boldsymbol{m}_{v_L}^{(\mathrm{GT})}(n)$.

1: Compute the Yul-Walker power spectrum density of $\boldsymbol{m}_L(n)$; $\boldsymbol{p}_L(f)$.

2: Calculate the corresponding frequency to the minimum of valley in $\boldsymbol{p}_L(f)$; f_d.

3: Decompose $\boldsymbol{m}_L(n)$ to J levels of approximation and form matrix $\boldsymbol{A}_L(n)$.

4: Derive the optimized matrix $\boldsymbol{w}_L(n)$ according to Eq. (4.9)

decomposition and the mother wavelet function which is deployed are discussed later in Section 4.2.4.

Once the reconstructed approximation matrix $\boldsymbol{A}_L(n)$ is computed, an unconstrained nonlinear optimization based on Quasi-Newton technique [237] is deployed to derive the adaptive weight vector $\boldsymbol{w}_L(n) \in \mathbb{R}^J$ such that the error $e_L(n)$ defined below is minimized

$$e_L(n) = \sqrt{\frac{1}{L}\big(\boldsymbol{A}_L(n)\boldsymbol{w}_L(n) - \boldsymbol{m}_{v_L}^{(\mathrm{GT})}(n)\big)^T\big(\boldsymbol{A}_L(n)\boldsymbol{w}_L(n) - \boldsymbol{m}_{v_L}^{(\mathrm{GT})}(n)\big)}. \tag{4.9}$$

This completes one iteration of the HPA scheme. Vector $\boldsymbol{w}_L(n)$, the cut-off frequency f_d, and the ground truth values $\boldsymbol{m}_{v_L}^{(\mathrm{GT})}(n)$ are the Hyper-parameters constituting output of the HPA scheme and are fed into the RTP scheme described next. Algorithm 9 summarizes the HPA scheme.

Real-time Tremor Prediction (RTP) Scheme

The RTP scheme performs real-time data analysis on the measurement signal $\boldsymbol{m}(n)$ at the current time instant n, and provides as an output a myopic predicted version of the tremor signal for the next time instant $(n+1)$. In this scheme, some predefined parameters are fixed and do not change over time, but some of the hyper-parameters are updated via the HPA scheme. The HPA scheme provides optimized hyper-parameters for the RTP to increase the overall real-time performance of the system for tremor extraction. In this scheme, the KF and wavelet decompositions are jointly incorporated to provide predictions for the tremor signal. Since the wavelet decomposition is employed in this scheme and the goal is to develop a real-time system, it is important to keep the amount of calculations as low as possible. The minimum length of a signal which is going to be decomposed with wavelet transforms, should be at least 2^J. Hence, the RTP scheme operates on the last 2^J samples of the measurement signal $\boldsymbol{m}_J(n)$ defined as follows

$$\boldsymbol{m}_J(n) \triangleq [m(n-2^J+1), \ldots, m(n)]^T. \tag{4.10}$$

Now, the matrix $\boldsymbol{A}_J$ should be incorporated in the KF formulations. Please note that for each time sample that the system receives one new measurement and n increases, the RTP scheme slides one sample ahead, and a new $\boldsymbol{m}_J(n)$ is formed. The independent variable k is also defined to demonstrate the calculations within the RTP scheme. At the first step, the signal $\boldsymbol{m}_J(n)$ is decomposed into J levels and J approximation signals are extracted. Then the approximations are put together to form the matrix $\boldsymbol{A}_J \in \mathbb{R}^{J\times 2^J}$. Note that the number of columns in matrix $\boldsymbol{A}_J$ (computed within the RTP scheme) is different from that of matrix $\boldsymbol{A}_L$ that is computed.

I propose to use Kalman filter to recursively iterate over the last 2^J samples to compute a predictive value for the approximations in the next coming iteration. For example, when the current iteration is k, samples $(n-2^J+1)$ to (n) are used to provide predictions for time iteration $n+1$. Index k is chosen to discriminate between the independent variable n that is used for the whole time domain of the problem. Note that k ranges from 1 to 2^J. Within the framework of KF, columns $\boldsymbol{a}_J(\cdot)$ of matrix $\boldsymbol{A}_J$ are defined as the state vector, i.e.,

$$\boldsymbol{x}(k) \triangleq \boldsymbol{a}_J(k). \tag{4.11}$$

The following linear state-space model is considered to represent the evolution of state vector over time

$$\boldsymbol{x}(k) = \boldsymbol{F}\boldsymbol{x}(k-1) + \boldsymbol{w}(k), \tag{4.12}$$

where matrix $\boldsymbol{F}$ is a diagonal matrix $\boldsymbol{I}_J$ of appropriate dimension. Observed signal values are considered as the measurement $z(k)$ in the KF recursions, i.e., $z(k) \triangleq m_J(k)$. The observation model is constructed based on the hyper-parameters (vectors $\boldsymbol{w}_L(n)$ and $\boldsymbol{m}_{v_L}^{(\mathrm{GT})}(n)$) which are provided by the HPA scheme. The observation model is set to the optimized weights $\boldsymbol{w}_L(n)$ reported by the HPA scheme. It is worth mentioning that the weight vector $\boldsymbol{w}_L(n)$ is updated whenever the HPA scheme re-runs; however, the KF matched to that window uses the same model during its operation. Therefore, index k is not required here. The observation noise $v(k)$ is considered to be a zero mean Gaussian distribution with variance R which is equal to the variance of the ground truth for tremor signal. The ground truth for tremor signal $(\boldsymbol{m}_{v_L}^{(\mathrm{GT})}(n))$ is considered as the known bias of the model. The observation model is, therefore, given by $z(k) = \boldsymbol{w}_L(n)^T\boldsymbol{x}(k) + m_{v_L}^{(\mathrm{GT})}(k) + v(k)$. In this context, the Kalman Filter recursions are used to provide myopic predictions as follows

$$\begin{aligned} \hat{\boldsymbol{x}}(k|k-1) &= \boldsymbol{F}\hat{\boldsymbol{x}}(k-1|k-1) && (4.13)\\ \boldsymbol{P}(k|k-1) &= \boldsymbol{F}\boldsymbol{P}(k-1|k-1)\boldsymbol{F}^H + \boldsymbol{Q}. && (4.14) \end{aligned}$$

By incorporation of new observation, the states are updated as

$$\begin{aligned} \boldsymbol{r}(k) &= \boldsymbol{z}(k) - \boldsymbol{w}_L(n)^T\hat{\boldsymbol{x}}(k|k-1) - m_{v_L}^{(\mathrm{GT})}(k) && (4.15)\\ \boldsymbol{S}(k) &= \boldsymbol{w}_L(n)^T\boldsymbol{P}(k|k-1)\boldsymbol{w}_L(n) + R. && (4.16)\\ \boldsymbol{K}(k) &= \boldsymbol{P}(k|k-1)\boldsymbol{w}_L(n)\boldsymbol{S}(k)^{-1} && (4.17)\\ \hat{\boldsymbol{x}}(k|k) &= \hat{\boldsymbol{x}}(k|k-1) + \boldsymbol{K}(k)\boldsymbol{r}(k) && (4.18)\\ \boldsymbol{P}(k|k) &= \boldsymbol{P}(k|k-1) - \boldsymbol{K}(k)\boldsymbol{w}_L^T(n)\boldsymbol{P}(k|k-1). && (4.19) \end{aligned}$$

Once iterations of the Kalman filter based on Eqs. (4.15)-(4.19) is completed based on observations $z(k)$, for $(1 \leq k \leq 2^J)$, the last updated estimate $\hat{\boldsymbol{x}}(k)$ is predicted one-step

Algorithm 10 THE RTP FRAMEWORK

Input: The measurement signal: $\boldsymbol{m}(n)$, and; The optimized weights; $\boldsymbol{w}_L(n)$.
Output: The predicted voluntary signal; $\hat{\boldsymbol{m}}_v(n)$
Segment the last J samples of the measurement signal and form $\boldsymbol{m}_J(n)$
Perform KF on $\boldsymbol{m}_J(k)$ based on Eqs. (4.11)-(4.19) for $k = 1, \cdots, 2^J$
Compute $\hat{\boldsymbol{m}}_v(n) = \boldsymbol{w}_L^T(n)\hat{\boldsymbol{x}}(k|k-1)$ for $k = 2^J + 1$ as the prediction for the next time instance.

Algorithm 11 THE OVERALL WAKE FRAMEWORK

Input: The measurement signal; $\boldsymbol{m}(n)$.
Output: The predicted voluntary signal; $\hat{\boldsymbol{m}}_v(n)$
Wait until L number of samples are ready.
Form $\boldsymbol{m}_L(n)$
$[\boldsymbol{w}_L,\ f_d] \ = \ \text{HPA}(\boldsymbol{m}_L(n))$
loop
  if $modulo(n/L) \in \mathbb{Z}$ **then**
    $[\boldsymbol{w}_L,\ f_d] \ = \ \text{HPA}(\boldsymbol{m}_L(n))$
  else
    Execute the RTP scheme:
    for $\{k = 1, ..., 2^J\}$ **do**
      Form $\boldsymbol{m}_J(n)$
      Perform KF on $\boldsymbol{m}_J(n)$ based on Eqs. (4.11)-(4.19).
    end for
    $\hat{\boldsymbol{m}}_v(n+1) = \boldsymbol{w}_L^T(n)\hat{\boldsymbol{x}}(k|k-1)|_{k=2^J+1}$
  end if
end loop

forward and $\hat{\boldsymbol{x}}(k|k-1)$ is used as the predicted state for next time sample which has not happened yet. Algorithm 10 summarizes different steps of the RTP scheme.

The Overall Workflow of WAKE

Algorithm 11 outlines the overall work-flow of the WAKE framework. As discussed earlier, the HPA scheme, operates when L new measurement samples are available. Hence, when the algorithm starts, the algorithm waits to receive the first L samples to adjust the hyper-parameters for the RTP scheme, then the RTP scheme starts operating. The RTP continues till L new samples are ready for the HPA.

4.2.3 Parameter Selection

As outlined in the previous section, the WAKE framework incorporates plenty of parameters and hyper-parameters to be identified, and this section focuses on comprehensive sets of validation to fine-tune such parameters. Please note that to evaluate the performance of system, the Normalized Root Mean Square Error (NRMSE) [81] is calculated as given below

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(\boldsymbol{m}_v^{(\mathrm{GT})}(n) - \hat{\boldsymbol{m}}_v(n)\right)^2}}{\max(\boldsymbol{m}_v^{(\mathrm{GT})}) - \min(\boldsymbol{m}_v^{(\mathrm{GT})})}. \tag{4.20}$$

Here, the ground truth ($\boldsymbol{m}_i^{(\mathrm{GT})}(n)$) for tremor signal is obtained by off-line sharp filtering of the measurement based on the mean of f_d over all of the executions of the HPA scheme. The predicted tremor ($\hat{\boldsymbol{m}}_i(n)$) is obtained by subtracting the predicted voluntary movement ($\hat{\boldsymbol{m}}_v(n)$) from the measurement signal.

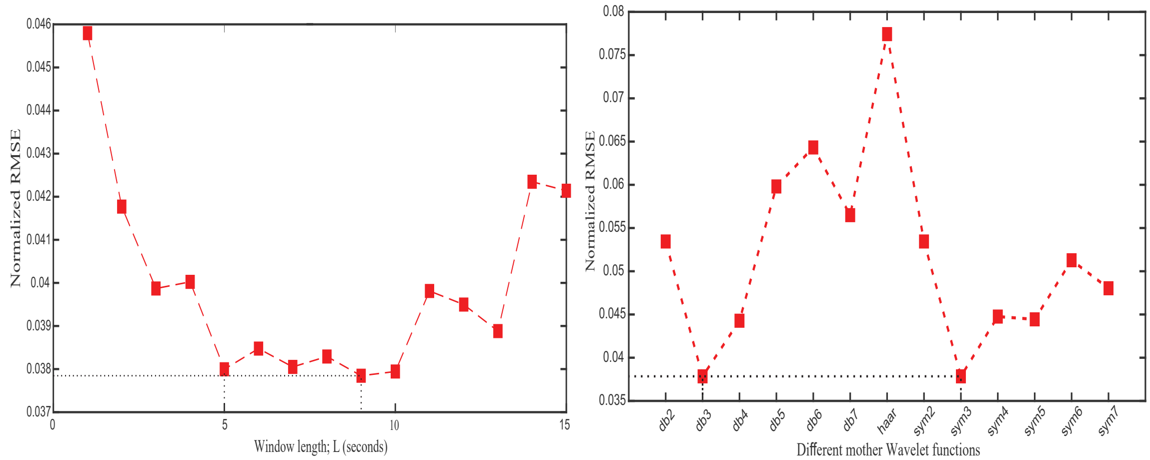
In the rest of this section, the results of several experiments performed to properly select the following design variables and parameters are presented: (i) The window length (L); (ii) Mother wavelet function and its effect on the overall performance; (iii) Level of decomposition in wavelet transform and its effects on the performance, and; (iv) The model-order (p in the Yul-Walker power spectrum estimation). To have a fair comparison, all the experiments are performed based on the same measurement signal with its Power Spectral Density (PSD) shown in Figure 4.2.

Effects of Window Length (L)

In order to select the best value for the window length, L, all the other parameters were set to a fixed value except for L, which was varied within the range of 1-15 seconds with steps of 1 second. Figure 4.3(a) illustrates the variation of the overall performance as a function of the changes in the window length L. From the results, it is observed that the minimum error (maximum achievable performance) occurs at $L = 9$ seconds which results in NRMSE of 0.0378. It is worth mentioning that the performance of the system at $L = 5$ seconds is rather close to that of the case with $L = 9$. To construct a more dynamic system for tracking the variation in behavior of the tremor signal with less delay, the window length was set equal to 5 seconds for the following experiments.

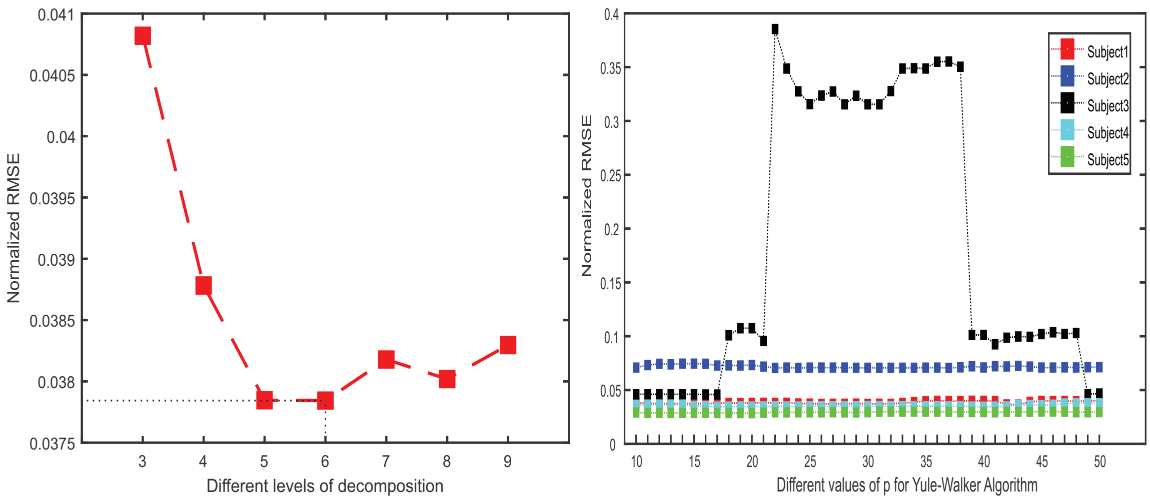
Effects of Mother Wavelet

Both the RTP and HPA schemes use the same mother wavelet function for their operations, therefore, the overall performance of the WAKE framework was assessed by investigating the following different functions including: Haar wavelet, db wavelets, and Symlets (sym) mother functions. As shown in Figure 4.3(b), the following mother functions are compared: *"db2", "db3", "db4", "db5", "db6", "db7", "haar", "sym2", "sym3", "sym4", "sym5", "sym6", "sym7"*. It is observed that the minimum NRMSE value is obtained over the *"db3"* and *"sym3"* mother functions, therefore, in the rest of the experiments, the "sym3" function is utilized.



(a) (b)

Figure 4.3: (a) NRMSE computed based on varying window length, L. (b) Effect of different mother wavelet functions in the overall performance.



(a) (b)

Figure 4.4: (a) Effect of different levels of decomposition in wavelet transforms in the overall performance. (b) Effect of parameter p in Yul-Walker method for 5 different subjects.

Effect of Different Levels of Decomposition (J)

I investigated effects of the decomposition levels (J) on the overall performance in terms of the achievable NRMSE. The level of decomposition directly influences the length of the sliding window in the RTP scheme, since the length of sliding window is equal to 2^J. The results of this experiment are depicted in Figure 4.4(a) where it is observed that the minimum NRMSE of 0.03784 is achieved with $J = 6$. We decided to set $J = 5$ as the NRMSE for $J = 5$ is 0.03785, which is fairly close to the minimum value while $J = 5$ levels of decomposition imposes less computational burden.

Table 4.1: Results of the BMFLC, E-BMFLC and WAKE frameworks on the synthetic data.

Signal	SNR_{in}	PRF%			SNR_{out}			NRMSE		
		BMFLC	E-BMFLC	WAKE	BMFLC	E-BMFLC	WAKE	BMFLC	E-BMFLC	WAKE
Signal1	7.5	11.97	12.84	**4.3**	9.22	8.91	**13.67**	0.12	0.13	**0.0734**
Signal2	5.1	15.94	16.41	**7.42**	**7.98**	7.85	7.42	0.14	0.14	**0.0964**
Signal3	0.3	31.49	30.72	**22.95**	5.02	5.13	**6.39**	**0.16**	0.2	0.1695
Signal4	-4	74.2	**67.89**	68.8	1.3	1.68	**1.72**	0.3	**0.29**	**0.2904**
Signal5	4.6	22.23	26.23	**8.13**	6.53	5.81	**10.9**	0.17	0.18	**0.1009**
Signal6	0.3	57.45	73.29	**45.56**	2.41	1.35	**3.32**	0.27	0.3	**0.2414**
Signal7	-4.9	183.18	240.61	**135.74**	-2.63	-3.81	**-1.23**	0.48	0.55	**0.4077**
Signal8	20.9	4.83	4.15	**0.32**	13.16	13.82	**25**	0.08	0.07	**0.02**
Signal9	15.1	5.43	4.66	**0.9**	12.65	13.32	**20.46**	0.08	0.08	**0.0335**
Signal10	10	7.36	6.25	**2.71**	11.33	12.04	**15.67**	0.1	0.09	**0.0583**

Effect of Model-order (p)

The model-order (p) is the order of the *AR* model which is fitted to the data, therefore, its variation has direct impact on the frequency band selection which in turn would impact the overall performance of the WAKE framework. Based on the experiments, it is observed that effect of the model-order is subject-dependent and in general, it does not follow any particular pattern. Figure 4.4(b) shows the NRMSE results as a function of the model-order based on 5 different subjects. By considering the fact that increasing the value of p will increase the amount of processing workload, it is set to $p = 17$ in order to maintain the generality of the algorithm for different subjects.

4.2.4 Simulation Results

In this section,the adaptive auto-adjustable WAKE framework is evaluated based on three different datasets: (i) A synthetic dataset (Section 4.2.4); (ii) A real-dataset containing recordings from patients with action tremor (Section 4.2.4), and; (iii) A second set of real data collected from patients with Parkinson's disease (Section 4.2.4). All the design variables and parameters are selected based on the results of the experiments discussed above. The results of the proposed method are compared with the results of two successful techniques in tremor estimation and extraction, which are BMFLC [126, 238] and E-BMFLC [81].

Synthetic Dataset

Initially, the performance of the WAKE framework is evaluated in comparison to its state-of-the-art counterparts based on a designed synthetic dataset. The introduced synthetic dataset is critically important and provides priceless insights as it allows us to precisely track and assess different aspects of the WAKE framework and observe the effects of different features. The synthetic dataset is designed in a way that embodies the following two main features of PHT: (i) In most cases, there is a clear distinction between the frequency band of voluntary movement and that of the tremor, and; (ii) Tremor happens in a certain and limited frequency band. The synthetic signal is constructed by combining three different signals as $\boldsymbol{s}(n) = \boldsymbol{s}_v(n) + \boldsymbol{s}_w(n) + \boldsymbol{s}_t(n)$, where $\boldsymbol{s}_v$ is a sinusoidal with a dominant amplitude (compared to the other two signals) with frequency of less than 1Hz representing the voluntary movement. The second signal $\boldsymbol{s}_w$ is an additive white noise occurring at all frequencies and is generated by setting a frequency resolution (frequency gap) and then adding sinusoidal with different frequencies between 0 to $F_s/2$ with steps equal to the frequency resolution. The amplitude of sinusoidal follow a uniform distribution with specified mean and variance. The third signal $\boldsymbol{s}_t(n)$ represents the tremor and its frequency band is limited to 6-14Hz to be consistent with the band limits specified in [126]. The tremor signal $\boldsymbol{s}_t(n)$ is generated following the same procedure used to generate the white noise. Finally, for both $\boldsymbol{s}_t(n)$ and $\boldsymbol{s}_w(n)$, a random phase is added to the constituting sinusoidal in order to prevent synchronization of the sinuses resulting in formation of "Sinc" waveforms. Based on the above procedure, ten test signals are constructed sampled at $F_s = 100$Hz and each representing 50 seconds of activity. Each of the ten constructed signals represents a different Signal to Noise Ratio (SNR) calculated as SNR $= 10 \times \log_{10}(\frac{P_{sig}}{P_{noise}})$. Note that in these calculations, $P_{noise} = P(s_w) + P(s_t)$ and $P_{sig} = P(s_v)$, where $P(\cdot)$ denotes the power of a signal. In this evaluation, beside the NRMSE and SNR, another metric referred to as Power Ratio Factor (PRF) is defined as

$$\text{PRF}(n) = \frac{(\boldsymbol{s}_v(n) - \hat{\boldsymbol{m}}_v(n))^2}{(\boldsymbol{s}_v(n))^2} \times 100, \tag{4.21}$$

which is used as an alternative measure to evaluate the performance of the proposed WAKE framework. The results are demonstrated in Table. 4.1. It is observed that the WAKE significantly outperforms its counterparts.

As a final note, we would like to point out that unlike the currently existing techniques, the proposed WAKE framework demonstrates a deterministic processing pipeline, which does not require any parameter and structure tuning based on the input data. The technique can even automatically finds the frequency range of interest. This significantly enhances the practicality of the proposed approach specially for clinical and biomedical settings, under which excessive parameter tuning and sensitivity to the choice of parameters are not acceptable. As a result, it can be mentioned that the proposed technique is not a data-driven processing framework which requires several training examples to fine-tune the parameters. Thus, a higher number of data points, would not provide a better model and it would not change the performance of the technique. However, higher number of data points can help to better judge the performance, as such the evaluation results based on two real datasets in addition to the synthetic dataset considered here are reported in the next section. In short, the results of all evaluations supports the benefit of the proposed framework. For the evaluation phase, we believe that the synthetic data used in this sub-section provides a robust assessment tool for evaluating the proposed WAKE framework. Synthetic dataset helps us to precisely track the changes and transformations applied to the three components of the synthetic signal which are the voluntary signal, the involuntary signal and the white noise. Thanks to this dataset, we can more reliably calculate the signal-to-noise ratio, and also it enables us to understand how differently the WAKE algorithm handles the involuntary movement and the white noise.

The Motus Dataset

The Motus dataset is recorded with bi-axial gyroscopes which are developed at Motus Bioengineering Incorporation, Benicia, CA and is publicly available [239]. The collected signals associated with different subjects with hand tremor are recorded with sampling frequency of 100 Hz. The subjects with action tremor are asked to perform pronation/supination with their hand, while the gyroscope, mounted on their hand, is recording the movement's signal. The recordings, therefore, contain information about the voluntary movement (pronation/supination), as well as information corresponding to the action tremor. The results of applying the proposed WAKE on the Motus dataset are presented in Table 4.2. Samples of the tremor signal and the extracted voluntary component are shown in

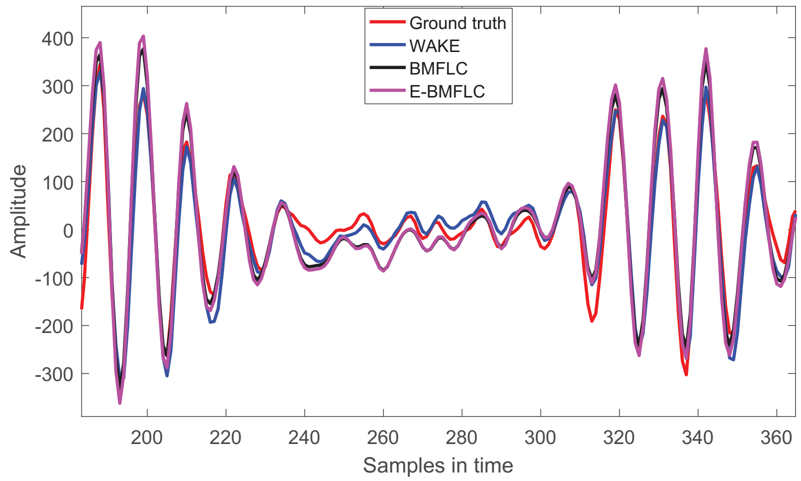


Figure 4.5: The predicted tremor derived by the proposed WAKE framework, the BMFLC method and the E-BMFLC method compared to the ground truth of the tremor derived by off-line processing of the signal (Subject 1 in Motus Dataset).

Table 4.2: Results of the application of the proposed method for Motus Dataset. The measure of error is Normalized-RMSE.

Subjects	BMFLC	E-BMFLC	WAKE
subject1	0.1130	0.0634	0.0385
subject2	0.1015	0.1263	0.0721
subject3	0.0927	0.0742	0.0444
subject4	0.1113	0.0622	0.0355
subject5	0.1043	0.0444	0.0279
Average	**0.1046**	**0.0741**	**0.0437**

Figures 4.5 and 4.6. As it is observed from the results, beside the fact that the WAKE framework provides real-time tremor extraction/estimation, it provides drastically improved results and significantly outperforms its well regarded counterparts.

As a side note to this discussion, we would like to also elaborate on better performance of the E-BMFLC in comparison to the BMFLC. In brief, the harmonic model used in the BMFLC is much smaller and different from the model used in the E-BMFLC. This means that in the BMFLC technique, the harmonic model only operates for the frequency range of tremor and just the harmonics for tremor are estimated and updated. In other words, the BMFLC does not utilize a general large harmonic model that models the whole signal. Instead, it tries to keep track of the pre-defined frequency band of the tremor. For example, for a tremor occurring in the range of 12 to 20 Hz, the BMFLC fits a model of harmonics from 12 and 20 Hz. On the contrary, the E-BMFLC models the range from 0 to 20 Hz and after estimating the parameters in each iteration, a part of the estimated harmonics will be considered as the tremor. Consequently, the E-BMFLC method is expected to better track the dynamics of the tremor in comparison to the BMFLC.

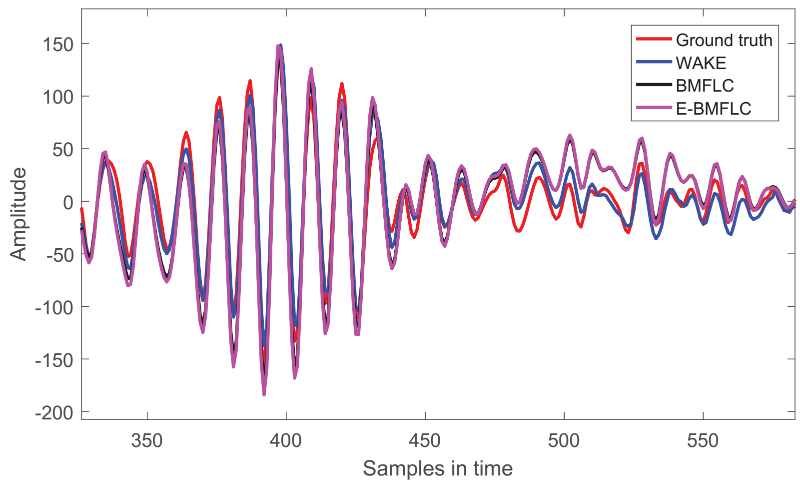


Figure 4.6: The predicted tremor derived by the proposed WAKE framework, the BMFLC method and the E-BMFLC method compared to the ground truth of the tremor derived by off-line processing of the signal (Subject 4 in Motus Dataset).

The Smartphone Dataset

To further elaborate on the generality and superior performance of the proposed WAKE framework, a third set of tremor recordings are employed from 10 patients with Parkinson's Disease. This dataset is provided by "*Hospital Clinic de Barcelona, Barcelona, Spain*" [168, 240]. The signals are recorded with the built-in tri-axial accelerometer of a smartphone (in this set iPhone 5S, Apple Inc., USA). The smartphone is mounted on the dorsum of the hand which shows more sever tremor. The signals in this dataset are recorded with the sampling frequency of 100 Hz, constituting of three channels of data, representing the acceleration in X-axis, Y-axis and Z-axis. The patients were asked to have their arms stretched while their upper limb is fully extended in front of them and their palms are facing the ground. Figures 4.7 and 4.8 provide demonstrating examples of the performance of the WAKE framework in extracting the tremor signal from the Smartphone dataset.

This data set which is collected by a different device and a different team is used to illustrate the efficacy of the proposed WAKE framework in comparison to the state-of-the-art techniques. As the measurement device for this dataset is easily accessible, this set is considered as a valuable development testbed. From clinical stand point, any diagnostic framework developed on this dataset would be practically more applicable as it does not require expensive and complex sensors. On the other hand and from signal processing stand point, the main difference between this dataset and the Motus one (utilized in Section 4.2.4) is the quality of the recordings. The Motus set is recorded with medical-grade sensors, specialized for tremor measurements, therefore, provides very clean and noise-free signals. The smartphone dataset, however, is recorded with the internal accelerometers of a smartphone, which are employed for general applications and thus the quality of the recorded signals is not as high, imposing a high degree of complexity to the processing pipeline to extract the voluntary movement. Consequently, the Smartphone dataset incorporated for performance evaluation can be considered as a difficult case study from signal processing

Table 4.3: Results of the BMFLC, E-BMFLC, and the proposed WAKE frameworks on the smartphone dataset. Term "*wname*" in the table denotes the mother Wavelet function.

Subject	x-axis *p=19, wname='db5' J=10, L=2^{10}*			y-axis *p=15, wname='db7' J=10, L=2^{10}*			z-axis *p=20, wname='db7' J=10, L=2^{10}*		
	BMFLC	E-BMFLC	WAKE	BMFLC	E-BMFLC	WAKE	BMFLC	E-BMFLC	WAKE
1	.1284	.0895	**.0657**	.1506	.1038	**.0873**	.096	.1	**.064**
2	.0919	**.0728**	.0862	.1102	**.0640**	.0811	.0713	.0519	**.0433**
3	.0596	**.0366**	.0610	.1621	.0958	**.0511**	.0419	**.0291**	.0325
4	.1093	.0767	**.0453**	.1112	.0746	**.0475**	.0754	.0599	**.0477**
5	.085	**.0648**	.078	.1015	.0574	**.0475**	.0844	.0528	**.0507**
6	.0637	.0638	**.0592**	.0647	.0391	**.0351**	.0661	.0427	**.0371**
7	.0881	.0499	**.0462**	.1892	.0877	**.0835**	.0815	.0501	**.0367**
8	.0629	**.0398**	.082	.1608	.0717	**.0349**	**.053**	.0716	.0914
9	**.0419**	.0477	.0431	.1081	.0665	**.0321**	.0648	.068	**.052**
10	.0801	.0626	**.0563**	.0688	**.0455**	.0564	.0775	.0438	**.0431**

point of view.

Table 4.3 provides the performance results obtained via the proposed WAKE framework, the BMFLC, and the E-BMFLC frameworks. It is worth mentioning that the same parameter selection process is performed for the smartphone dataset, separately for each channel of data. Hence, the information in *italic* font presented in Table 4.3 denote the best hyper-parameters for each axis. For parameter-selection, the signals from the first subject are analyzed and then the parameters are employed for other subjects. The numbers in the table represent the NRMSE between the predicted tremor and the ground truth signal for tremor. Figures 4.7 and 4.8 illustrate the performance of the proposed WAKE framework compared to its counterparts including the BMFLC method and the E-BMFLC method. It is observed that the results obtained from the proposed framework are significantly improved in comparison to its counterparts. As stated previously, the smartphone dataset is a difficult case study from signal processing point of view, being capable of significantly outperforming the state-of-the-art on this dataset illustrates superiority of the proposed WAKE framework. This evaluation clearly shows the superior performance of the proposed WAKE approach and supports its practicality and reproducibility for signals collected with different machines.

4.3 The PHTNet Framework

Besides concerns regarding the computational power and capacity of existing frameworks for an ultimate predictive model, there is a need for characterizing tremor based on a sizable inclusive dataset, that covers possible pathological variations causing diverse types of tremor signals in terms of spectrotemporal behavior, dynamic nature, temporal dependencies, and sub-movements. Without such a data atlas, conservative and impractical assumptions would be considered to define a ground truth reference for designing and validating the techniques. We argue that due to the high degree of inter- and intra-subject variability of tremor characteristics, the solutions designed and validated based on a limited dataset may not be generalizable for translation to the clinic and ADL for PD and ET patients.

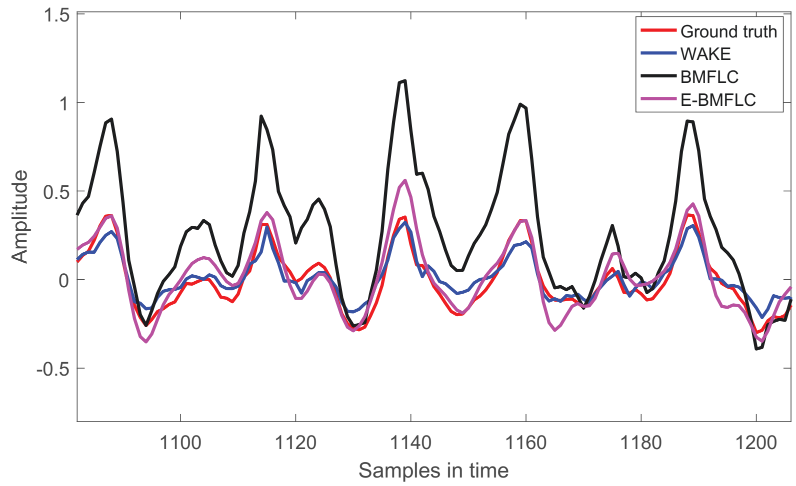


Figure 4.7: The predicted tremor derived by the proposed WAKE framework, the BMFLC method and the E-BMFLC method compared to the ground truth of the tremor derived by off-line processing of the signal (Subject 1 - X-axis in Smartphone Dataset).

Thus, building a representative and sizable dataset coupled with designing a predictive model with high spectrotemporal capacity are critical for developing a PHT removal and prediction framework that is robust to inter- and intra-subject variability of tremor characteristics. This unmet need is addressed and discussed in this section by utilizing a novel data-driven deep neural network modeling technique with a unique tremor extraction capacity augmented by predictive power and trained based on our unique dataset.

The dataset includes kinematic motion recordings of 81 PD and ET patients (87.5 hours of recordings) collected in a movement disorders research laboratory based on a rigorous systematic protocol. This is the largest dataset known to date permitting the generation of a strong high-capacity model. Developing a PHT extraction framework by employing large datasets, which include recordings of an extensive number of patients over long periods of time, takes a significant range of possible variations in the characteristics of tremor into account, adds more generalization to the framework, and makes the technique adaptable to inter and intra-patient variabilities, nonstationary and nonlinear behaviors of tremor signals. In other words, utilization of such larger dataset allows the network to more securely avoid the curse of overfitting to the training samples by observing a more diverse range of possibilities in the PHT behavior. The proposed model, referred to as PHTNet, is a real-time and assumption-free neural model developed based on bidirectional recurrent neural networks. PHTNet can process measurement signals in both online and offline fashion, and provides one-sample-ahead (one sample ahead) predictions on the voluntary component of hand motion signals, which is the ultimate temporal resolution for this application. It should be noted that the time resolution of signal processing plays a very important role when the control loop gain and the frequency of activations in a control system are high (such as the robotic rehabilitation devices) [241]. Therefore, one sample ahead-of-time prediction can play a very important role for compensation of external disturbances and achieving the control goals. On the other hand, a major difference between the proposed

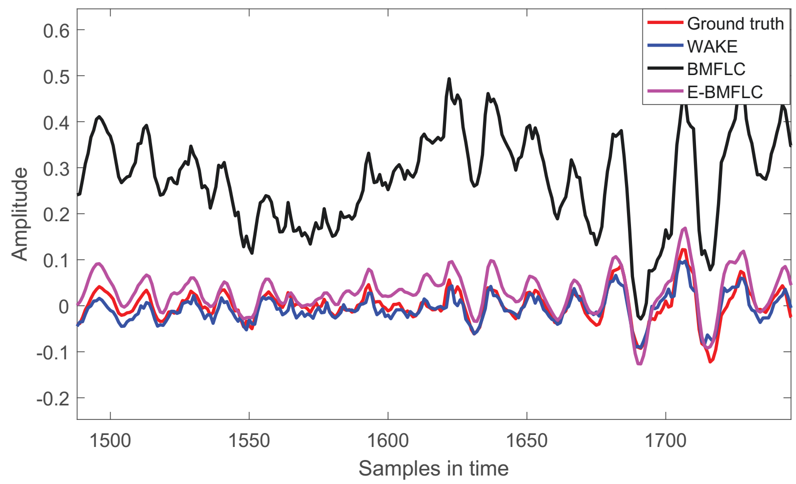


Figure 4.8: The predicted tremor derived by the proposed WAKE framework, the BMFLC method and the E-BMFLC method compared to the ground truth of the tremor derived by off-line processing of the signal (Subject 9 - Y-axis in Smartphone Dataset).

intelligent data-driven tremor extraction model and conventional filters is that since it is supported by the deep modeled connectivity in the dataset, it not only can remove phase latency but also has a one sample prediction. It is worth noting that phase lag (even in the order of 10 ms) can easily make a high-gain rehabilitation robot unstable which can sacrifice safety [81, 241–244]. Therefore, the proposed PHTNet, which not only removes the lag but also enhances the time resolution, provides a significant phase benefit, which is imperative for the control algorithm of rehabilitation and assistive technologies.

To address the need for a valid ground truth of the voluntary component of action tremor signals when training the PHTNet, through a novel design, a mixture of synthesized voluntary components and recorded rest and postural tremor signals from PD and ET patients are employed to teach the network to distinguish between the two motor behaviors. It is worth noting that this study is not made based on any potential differences between the characteristics of tremor in PD and ET. The data from two population of patients is collected to get a large variety of tremor characteristics and to empower our algorithm for a large population of users. One of the differences between PHTNet and previous works in the literature is that the characteristics of PHT are not bounded into a few commonly used parameterized assumptions in the spectral domain, as this is a highly dynamic feature. Instead, a data-driven approach, i.e., PHTNet, is employed, which learns how to internally evaluate the separation criteria and employ them for extraction and prediction of the voluntary component through several training samples, without explicitly introducing any characteristic of or assumptions on the tremor. Moreover, the devised training strategy of PHTNet teaches the network to model the nonlinear short-term and long-term temporal dependencies and minimize the error between the output sequence and the forthcoming samples of the ground truth signal to equip PHTNet with predictive behavior, which, for the first time, presents an ultimate temporal resolution.

It is worth noting that the PHTNet is not proposed as a tremor treatment procedure, rather, it is developed to enhance the efficacy of current tremor management methodologies and also improve the quality of services delivered to the patients by robotic assistive devices. In other words, PHTNet can be put into practice to extract the voluntary and involuntary components of action tremor signals with a high spatiotemporal resolution to help the neurologists with objectively assessing the characteristics of tremor over the course of action and time, needed to specify the dosage and plan of prescribed medications. In addition, PHTNet can be directly employed by robotic assistive and rehabilitative technologies to enhance the quality of delivered assistive and rehabilitative services by maximizing the performance in reducing the tremor and stabilizing the motion, and by minimizing the risk of amplifying the tremor component by the device. Failure in precisely removing the tremor component in the input signals to the assistive devices can result in abrupt and unpredictable force profiles generated by the device, which can sacrifice safety. Details can be found in our previous papers in [81, 245].

4.3.1 Methods

This section describes the basics of the proposed PHTNet, as well as the systematic data collection strategy employed to design and train the network.

Dataset

This dataset was collected from 81 PD and ET individuals who participated in a single-centre. The study protocol was approved by the Western University's Health Sciences Research Ethics Board (REB#: 104584 and 107433) at the London Movement Disorders Centre in London, Ontario, Canada. The study protocol is registered with the "*www.clinicaltrials.gov*" registry (Identifiers: NCT02551848 and NCT02668497). All experiments were conducted in accordance with the Declaration of Helsinki, as well as the Tri-Council Policy Statement of Ethical Conduct for Research Involving Humans in Canada. The ethics committee provided full board approval for this study protocol and the consent procedure was approved as required in the documentation checklist, submitted with the full study protocol. Demographics data of the PD and ET group are tabulated in the supplementary material. All participants were recruited through the Movement Disorder Centre, at the University Hospital, London, Ontario, Canada. All participants provided written informed consent regarding their participation in the study. The participants recruited met the inclusion/exclusion criteria [114, 118]. First participant's first visit and last participant's last visit occurred in March 2014 and January 2018, respectively.

A convenience sampling of 119 PD and 131 ET upper-limb tremor assessments were utilized to develop PHTNet. The PD group included 47 patients, 8 females and 39 males, with an average age of 71.51 ± 7.63, where 26 of them were de novo patients. 14 and 35 patients were recorded bilaterally and unilaterally, respectively. 45 patients were assessed in two sessions with a time interval of 6 weeks and only 2 patients participated once. The ET group included 34 patients, 13 females and 21 males, with an average age of 69.8 ± 6.12. This group included 22 de novo patients and the whole ET group was assessed bilaterally; 3 patients participated only once and the rest were assessed twice, with a time interval of 6 weeks.

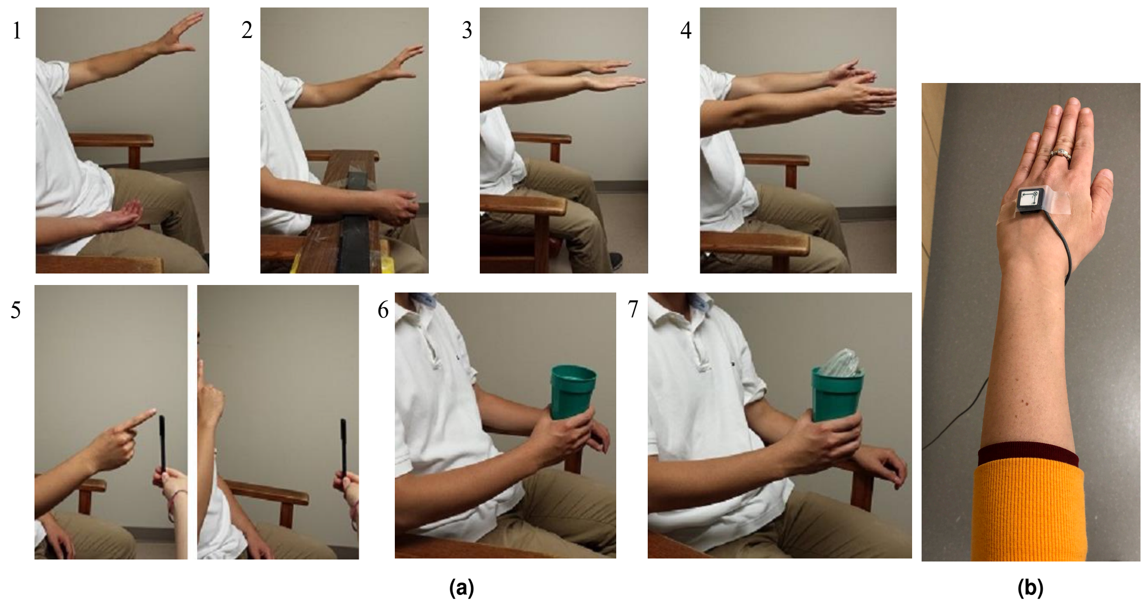


Figure 4.9: **(a)** Illustration of the 7 scripted tasks performed by PD and ET patients for each tremor assessment. Please note that, these are representative pictures (not including any patients). 1) Rest-1; 2) Rest-2; 3) Posture-1; 4) Posture-2; 5) Action tremor (repetitive finger to nose motion); 6) Load-1 (empty cup); 7) Load-2 (1-lb weight in the cup). **(b)** Placement of the 3-axis accelerometer sensor on the dorsum of a hand.

Kinematic analysis of upper-limb tremor was conducted by having participants perform a series of seven scripted tasks each held for 20 seconds over three trials, as previously described [114, 118] and illustrated in Figure 4.9a: two rest positions with the forearm supported on the lap ("Rest-1") or supported on a board ("Rest-2"), two postural positions with the arms pronated outstretched with palms facing downwards ("Posture-1") or with arms outstretched and palms facing each other ("Posture-2"), two weight-bearing tasks with participants holding an empty cup ("Load-1") and holding a cup with a 1-lb weight ("Load-2"), and one kinetic/action task where participants conducted a repetitive finger-to-nose action. Thus, 6 of the 7 tasks captured PHT in a static position (denoted as "static tremor") and the finger-to-nose dynamic task provided "action tremor" data. An inline 3D accelerometer sensor (#317A Noraxon U.S.A Inc.) was placed on the back of the hand, as illustrated in Figure 4.9b, to capture hand tremor in real-time using TeleMyoTM 2400T G2 at 1500Hz and transmitted to a computer running MyoResearch XP Version 1.08.0951,62. In total, 87.5 hours of data was used in this work collected from 81 patients (3 channels for each patient, 7 minutes per assessment, and 250 tremor assessments in total).

Data Preparation

To prepare the dataset for development of the PHTNet, all the recorded signals were down-sampled to 100 Hz. Based on the Nyquist sampling theorem [246], for full reconstruction of a sampled signal, one needs to set the sampling frequency at least to double that of the

Table 4.4: Categorization of the data for training, validation and testing purposes.

81 Patients (250 Tremor Assessments)		
20% Test Data (16 subjects)	20% Validation Data (16 subjects)	60% Training Data (49 subjects)
• Static tremor was used for quantitative evaluation	• Static tremor was used for validation of the model	• Static tremor was used for training the network
• The action tremor of the three sets was employed to qualitatively monitor the performance of the network.		

maximum meaningful frequency of the signal. It is worth noting that although downsampling may generally impose distortions on the signals, in the case of tremor removal with the frequency of interest generally being less than 15 Hz, downsampling to 100 Hz would impose minimum to no distortion to the spectral range of interest in the hand motion signals, while at the same time it avoids imposing excessive computational costs on the system. Thereafter, the dataset was divided into three sets for training, validation, and testing of the network. Table 4.4 explains the categorization of the data used for training and evaluation of PHTNet. It is worth mentioning that to impose harsh evaluation conditions on the PHTNet and strictly avoid leakage of information, directly or indirectly, from training set to the validation and test sets, the training, validation, and test sets are formed based on subjects. In other words, the recordings of 49, 16, and 16 subjects are respectively employed to form each of the development sets. Furthermore, due to the availability of a large dataset for development of PHTNet, three sets for training, validation, and testing are formed. Based on the aforementioned motivations in employing pseudo-synthesized data, the need for a valid ground truth in action tremor signals is addressed by synthesizing the voluntary component and mixing it with static tremor signals. The voluntary component was a sinusoidal signal with random amplitude, frequency, and phase, which was modeled as

$$m_v^{(\mathrm{GT})}(n) = a \sin(2\pi f t + \phi), \tag{4.22}$$

where amplitude, frequency and phase follow uniform distributions, i.e., $a \sim \mathbf{U}(0, 0.25)$, $f \sim \mathbf{U}(0,3)$Hz, and $\phi \sim \mathbf{U}(0, \pi)$, respectively. Since an additive model is assumed for the voluntary and involuntary components to build the motion signal, the synthesized voluntary component was mixed with the experimentally-collected static tremor signals from PD and ET participants, which were scaled to the range of $[0, 0.5]$, and PHTNet was fed with a pseudo-synthesized action tremor signal in the range of $[0, 1]$. Moreover, in this model (Eq. (4.22)), the frequency contents of the voluntary motion are assumed to spread over the range of $[0, 3]$ Hz, which is a reasonable assumption as very fast hand motions are not expected from the ET and PD participants, due to the rigidity and stiffness of the muscles. It is worth noting that this assumption is with regard to the voluntary component and does not impose any assumption on the involuntary component. Moreover, the spectral range for the pseudo-synthesized voluntary hand motion, compared to conventional methods, is a more relaxed assumption as typically this range is taken to be up to 1 Hz. Although the spectral range of $[0, 3]$ Hz takes a wide variety of motions into account, it does not imply that no tremulous activity occurs in this range, and this could be marked as one of the main advantages of this work over conventional methods.

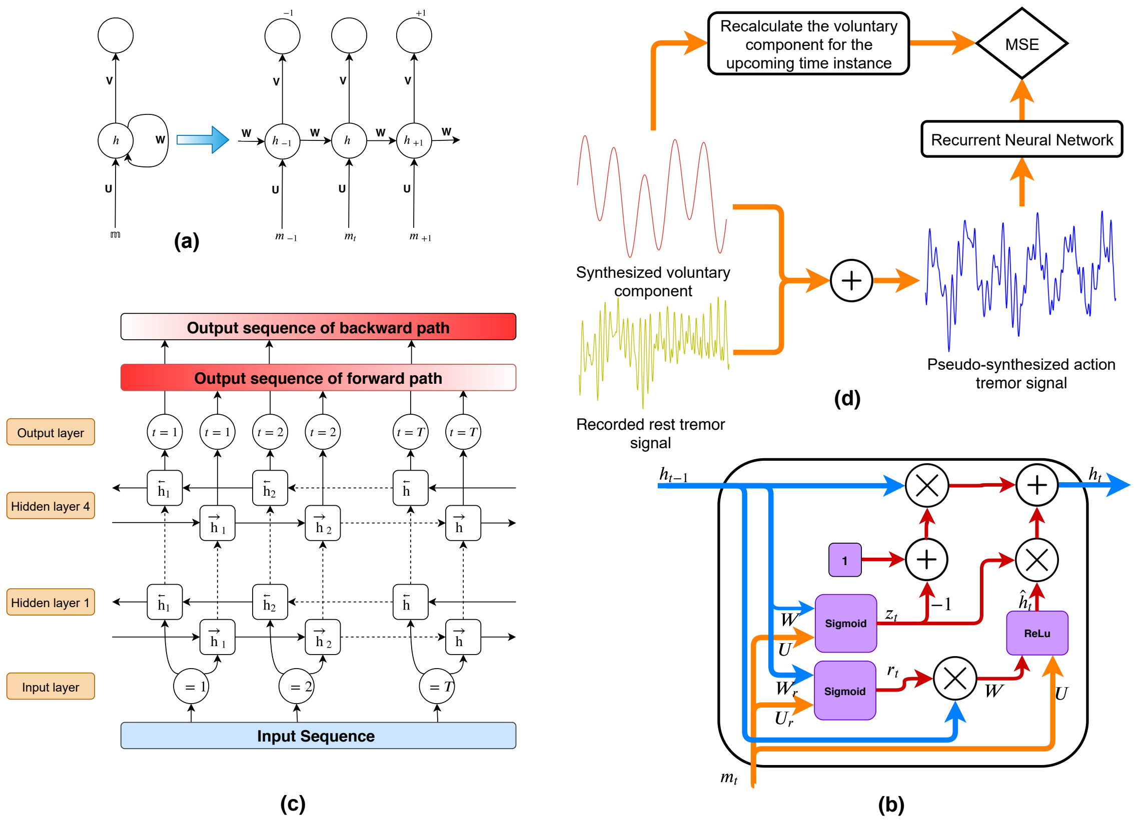


Figure 4.10: **(a)** A schematic of a one-layer RNN within the PHTNet with the unfolded version demonstrated on the right-hand side, which clearly shows the processing pipeline for different time instances. For the schematic on the left-hand side, it should be noted that the branch denoted by the weight $\boldsymbol{W}$ also applies one sample delay in time. **(b)** A gated recurrent unit (GRU) which is employed in PHTNet as the recurrent cell and is equipped with reset gate (r) and update gate (z). **(c)** The architecture of PHTNet, which is a 4-layer deep bidirectional recurrent neural network. $\overleftarrow{h}$ defines the backward cells for offline tremor elimination, and $\overrightarrow{h}$ defines the forward cell of the network for online tremor estimation/prediction. As shown in the diagram, the forward path is completely distinct from the backward path and their outputs are not merged into a single output sequence. The red gradient in the output blocks represents the degree of error in the extracted voluntary component. The high intensity of the red color symbolizes a high degree of error and the opposite mimics lower error rates. **(d)** The overall workflow of the proposed framework. Note that the voluntary component is recalculated for the next time instance and then is compared with the output of the network. This strategy is taken to enable the network with predictive features.

4.3.2 Internal Architecture of the PHTNet

Machine learning is defined as a study of statistical and mathematical models, which enable a computer to capture the behavior of a certain phenomenon without explicit instructions. Conventional machine learning methods are based on hand-crafted and user-engineered techniques developed to transform and represent raw data in a format which is perceivable by mostly-linear, or linear-in-parameter mathematical models. Performance of traditional machine learning methods, however, is normally restricted due to their limited modeling/learning capability. In addition, conventional machine learning methods require domain expertise and careful system design in order to have an acceptable performance. Therefore, representation learning methods [169] are introduced and developed such that the intrinsic patterns of input data are automatically inferred and extracted.

Recurrent Neural Network (RNN) models are a subcategory of representation learning methods [172] which are specialized in analyzing sequential data and detecting long-term and short-term temporal dependencies in signals based on nonlinear embedded memory. An RNN model consists of a sequence of hidden cells employed to process a stream of data. In RNN models, at each time instance, a combination of input sequence, i.e., hand motion signal, and hidden state vector of the previous time instance are analyzed together to update the state vector and pass it to the next time instance. This process continues until the whole sequence of hand motion measurements is analyzed and a meaningful representation is formed. RNNs have various designs to fit different applications, e.g., sequence to sequence RNNs are employed for machine translation tasks, and sequence to single-output RNNs are employed for classification tasks. Since in this work, a tremor-contaminated sequence is translated into to a clean sequence representing the voluntary component of hand motion, the sequence-to-sequence architecture for RNNs is employed. A typical RNN representation is depicted in Figure 4.10a and the formulations governing the RNN are given by

$$\boldsymbol{h}(k) = \mathrm{f}\big(\boldsymbol{b}+\boldsymbol{W}\boldsymbol{h}(t-1)+\boldsymbol{U}\boldsymbol{m}(t_1:t)\big), \tag{4.23}$$

and

$$\hat{\boldsymbol{y}}(k) = \mathrm{softmax}\big(\boldsymbol{c}+\boldsymbol{V}\boldsymbol{h}(k)\big), \tag{4.24}$$

where $\boldsymbol{m}(t_1:t) = [m(t_1),\ldots,m(t)]^T$ is the hand motion signal from time ($t_1 < t$) to time t as the input sequence of the network; $\boldsymbol{h}(k)$ is the hidden feature vector; $\boldsymbol{b}$ is the bias vector for the input nodes; $\boldsymbol{W}$ is the weight matrix for hidden-to-hidden connections; $\boldsymbol{U}$ denotes the input-to-hidden weights of the RNN; $\boldsymbol{c}$ is the bias vector for the output nodes; $\boldsymbol{V}$ denotes the weight matrix for hidden-to-output connections; and $\mathrm{f}(\cdot)$ denotes a nonlinear function; here the Rectified Linear Unit (ReLu) [247] is employed. Please note that the weights and biases in Eqs. (4.23)-(4.24) are derived/optimized during the training phase. The schematic of a GRU cell is shown in Figure 4.10b.

Using an RNN it can be expected that the output of the network for the very initial input samples is inaccurate and as the information propagates across the network and more samples of the input sequence are analyzed, the output becomes more accurate. Consequently, the output sequence becomes more reliable (in terms of its similarity to the ground truth signal) after a transient phase of initial inputs. As in this work, our goal is to develop an online and offline tremor extraction framework, the processing pipeline is structured in a bidirectional format which employs two parallel sets of recurrent cells for the two processing schemes. As shown in Figure 4.10c, forward cells are employed for online (predictive) processing of the input sequence where the maximum accuracy of estimation for the last samples of the output sequence is needed. Backward cells, on the other hand, are employed for offline processing of measurement signals, following the same logic for the forward cells. It is worth mentioning that in the utilized architecture, the base of which has been named in the literature as BRNN [173], the forward and backward hidden cells are usually followed by a mixing matrix which merges the outputs of the two paths. However, in this work, through an architectural modification of the model, the BRNN kernel is applied without the mixing matrix. This is done since the ultimate goal is to have two separate processing pipelines for both online and offline applications. Finally, Figure 4.10d, shows the devised training strategy to teach the network how to estimate/predict the voluntary motion.

A common problem with the classical versions of the RNN model described in Eqs. (4.23)-(4.24) is its weakness in capturing long-term patterns of the input sequence. This shortcoming is pronounced when long sequences of data need to be processed or when the input sequence encapsulates nonstationary patterns, which is the case for PHT extraction. Moreover, training of these networks is very critical since the problems of vanishing or exploding gradients are prevalent. To address these issues, two gates namely "reset gate" and "update gate" were adopted from the literature and integrated into the conventional hidden cells, and GRU cells [174] were developed. The reset gate determines the degree of dismissing old information and considering the data from input in the current time. The update gate, on the other hand, defines the degree of updating a hidden state based on the newly arrived data [175]. Therefore, the Eq. (2.3) is updated as

$$\boldsymbol{r} = \sigma\big(\boldsymbol{U}_r \boldsymbol{m}(t_1:t) + \boldsymbol{W}_r \boldsymbol{h}(t-1)\big), \tag{4.25}$$

$$z = \sigma\big(\boldsymbol{U}_z \boldsymbol{m}(t_1:t) + \boldsymbol{W}_z \boldsymbol{h}(t-1)\big), \tag{4.26}$$

$$\tilde{\boldsymbol{h}}(k) = \mathrm{ReLU}\big(\boldsymbol{U}\boldsymbol{m}(t_1:t) + \boldsymbol{W}(\boldsymbol{r} \odot \boldsymbol{h}(t-1))\big), \tag{4.27}$$

and $$\boldsymbol{h}(k) = (1-z)\boldsymbol{h}(t-1) + z\tilde{\boldsymbol{h}}(k), \tag{4.28}$$

where the reset gate is denoted by r and the update gate is denoted by z. Consequently, their corresponding weights are denoted by $\boldsymbol{U}_r, \boldsymbol{W}_r$ and $\boldsymbol{U}_z, \boldsymbol{W}_z$, respectively. The term σ denotes a logistic sigmoid function. As PHT can be a highly dynamic and nonstationary phenomenon, GRU cells are utilized in this work to better capture the long-term behavioral variations of the hand motion signal.

To conclude this section, a modified deep BRNN is utilized to process the hand motion signals and to estimate and predict the voluntary motion of patients. It should be added that, deep learning methods, which are a subcategory of representation learning techniques, are composed of several levels of simple but nonlinear units. Each level transforms and abstracts the raw input to a point where complex functions are learned [169]. The proposed PHTNet is a deep architecture constructed by stacking four BRNN layers such that the output of one layer is provided as the input to the next layer.

4.3.3 Proposed Geometry of the PHTNet

Rigorous performance validation of PHTNet is satisfied by grid-searching over potential hyperparameters of the network, i.e., the length of input sequence, the number of RNN features, the number of hidden layers in the deep architecture, and the learning rate for the optimization algorithm. To identify the number of hidden layers in the RNN architecture, a comprehensive grid-search approach is taken to compare the MSE value over validation and test sets across different number of hidden layers. In this regard, the error of network over validation and test sets is obtained and plotted as a function of the number of hidden layers. As shown in Figure 4.11, the best performance of the PHTNet is achieved when 4 hidden layers are stacked to each other. In PHTNet, 4 GRU cells are used to process the input sequence. To select the length of the input signal to be fed to the PHTNet with the aim of maximizing the overall performance of the network, a comprehensive grid-search approach is conducted to evaluate and compare effects of using different lengths of the input signals. To this aim, the performance of PHTNet is investigated in terms of normalized Mean Squared Error (MSE) over $24,300$ validation samples in 5 cases, where the input signal length is set to 1, 2, 3, 4, and 5 seconds for each case. The results of this experiment are shown in Figure 4.12. As it can be observed, performance of the PHTNet improves as the length of the input signal increases to a certain point and then either degrades or remains, more or less, unchanged. It is worth mentioning that while the performance of PHTNet remains almost a constant value by increasing the length of the input signal beyond 4 seconds, the computational costs of running the algorithm will increase. The PHTNet, therefore, yields the best performance when input sequences of 4 seconds are fed into the network considering jointly accuracy and computational cost in perspective. Moreover, and based on our rigorous validation procedure, it turned out that using 400 features for $\boldsymbol{h}(t)$ in Eq. (4.28) best abstracts and represents the motion signal in terms of providing maximum estimation accuracy of the tremor component. The network is trained based on minimizing the MSE value, and the ADAM Optimizer [248] with a learning rate of 0.0001 employed for this purpose (the learning rate defines the degree of update for the parameters of a neural network in the training session).

As stated previously, providing predictions on the estimated voluntary component of hand motion signals is an important feature for robotic rehabilitation technologies and an unmet need in the literature. A contributing factor for the absence of this feature is the highly dynamic behavior of PHT in and across affected individuals. Predictive operation of a PHT elimination framework grants the robotic systems enough time to adjust their parameters for the subsequent tremulous events. To address this, a modified scheme in feeding the network with training examples is proposed in the thesis. In fact, instead of normally feeding PHTNet with $\boldsymbol{m}(t_1:t)$ and calculating the MSE value between $\hat{\boldsymbol{y}}(t_1:t)$ and $\boldsymbol{m}_v^{(\mathrm{GT})}(t_1:t)$, the estimation error is measured between $\hat{\boldsymbol{y}}(t_1:t)$ and $\boldsymbol{m}_v^{(\mathrm{GT})}(t_1+1:t+1)$ to train PHTNet. In other words, and as depicted in Figure 4.10d, our devised strategy is to minimize the error between the network's output and a shifted version of the ground truth signal, which teaches the network how to estimate the voluntary component of hand motion and also to predict the upcoming future samples of the voluntary component. It is worth emphasizing that to enable the PHTNet with predictive behavior, no translational parameter or hyper-parameter is considered and this feature is taught to the network by our devised training strategy. In other word, this behavior becomes an intrinsic characteristic of the PHTNet. For this, when the network is trained to minimize the error between the input sequence and the advanced-in-time output sequence, it is actually learning how to predict the voluntary action in time. To clearly itemize the step-by-step training procedure of the PHTNet, its algorithmic workflow is summarized in Algorithm 12. Please note that the PHTNet could be employed as a plug-and-play model in practical applications and the itemized steps in Algorithm 12 only describe the development phase of the PHTNet.

Algorithm 12 The algorithmic overview of the development phase of the PHTNet

Require: The dataset of tremor assessments from 81 patients

Form the training, validation, and test sets by splitting the patients into three groups.
Extract segments of 4 seconds length from static tremor data, $\{\boldsymbol{s}_i\ ;\ i=1,\cdots,N\}$.
Produce synthetic voluntary component for each segment, $\{\boldsymbol{m}_{v_i}^{GT};\ i=1,\cdots,N\}$.
——————— In the training phase ———————
Form the set of $\{\boldsymbol{s}_i,\ \boldsymbol{m}_{v_i}^{GT}\ ;\ i\in$ Training set$\}$.
for Each pair of static tremor and synthetic voluntary component: **do**
    Generate action tremor data ($\boldsymbol{m}_i(t_1:t)$) by summing up $\boldsymbol{s}_i$ with $\boldsymbol{m}_{v_i}^{GT}$.
    Feed PHTNet with $\boldsymbol{m}_i(t_1:t)$ and get the output, $\hat{\boldsymbol{y}}_i(t_1:t)$.
    Calculate the error between $\boldsymbol{m}_{v_i}^{GT}$ and $\hat{\boldsymbol{y}}_i(t_1:t)$ according to Eq. (4.29)
    Update the parameters of PHTNet to minimize the error through backpropagation of error.
end for
——————— In the validation phase ———————
Form the set of $\{\boldsymbol{s}_i,\ \boldsymbol{m}_{v_i}^{GT}\ ;\ i\in$ Validation set$\}$.
for Each pair of static tremor and synthetic voluntary component: **do**
    Generate action tremor data ($\boldsymbol{m}_i(t_1:t)$) by summing up $\boldsymbol{s}_i$ with $\boldsymbol{m}_{v_i}^{GT}$.
    Feed PHTNet with $\boldsymbol{m}_i(t_1:t)$ and get the output, $\hat{\boldsymbol{y}}_i(t_1:t)$.
    Calculate the error between $\boldsymbol{m}_{v_i}^{GT}$ and $\hat{\boldsymbol{y}}_i(t_1:t)$ according to Eq. (4.29)
end for
Calculate the mean of the measured errors for the whole validation set.
Monitor the mean error of validation set after each training session to decide whether to continue training the network and also whether to keep the hyper-parameters of the network or update them with different values.
In case of stopping the training session, the measured mean error is the quantitative evaluation error of the PHTNet for the validation set.
——————— In the test phase ———————
Form the set of $\{\boldsymbol{s}_i,\ \boldsymbol{m}_{v_i}^{GT}\ ;\ i\in$ Test set$\}$.
for Each pair of static tremor and synthetic voluntary component: **do**
    Generate action tremor data ($\boldsymbol{m}_i(t_1:t)$) by summing up $\boldsymbol{s}_i$ with $\boldsymbol{m}_{v_i}^{GT}$.
    Feed PHTNet with $\boldsymbol{m}_i(t_1:t)$ and get the output, $\hat{\boldsymbol{y}}_i(t_1:t)$.
    Calculate the error between $\boldsymbol{m}_{v_i}^{GT}$ and $\hat{\boldsymbol{y}}_i(t_1:t)$ according to Eq. (4.29)
end for
Calculate the mean of the measured errors for the whole test set.
The measured mean error is the quantitative evaluation error of the PHTNet for test set.

4.3.4 Evaluation Metrics

We argue that the existing techniques for PHT elimination may suffer from the absence of a generalizable and inclusive method of extracting the ground truth for the voluntary component of action tremor to reliably measure the performance of the system. Hence, instead of employing conventional methods to extract the ground truth and calculate the performance, an inverse evaluation method is implemented. To this end, pseudo-synthesized

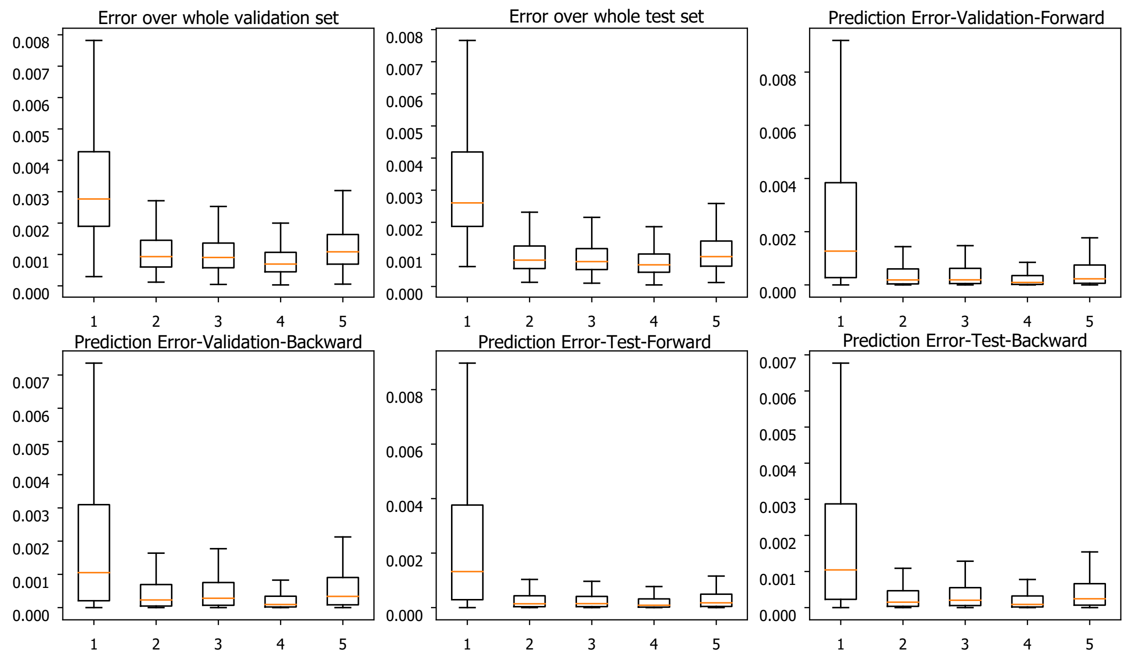


Figure 4.11: Comparison of normalized MSE (y-axis) over validation and test sets for different number of hidden layers (x-axis). Performance of the PHTNet in each test case is shown in box-plots, where the orange line indicates the median, the box indicates the range between 25% to 75% quartiles, and the lines indicate the standard deviation range. "Prediction Error" indicates the error for only the last sample of estimated voluntary signal. "Forward" and "Backward" indicate which path of the PHTNet is used.

action tremor signals are generated by mixing real static tremor recordings with a synthesized atlas of voluntary components, which provides an opportunity to numerically assess the performance measurement of PHTNet in training, validation, and test phases, without specifying assumptions to extract the ground truth signal. Existing methods on PHT elimination show limited performance in estimating the voluntary component of motion signals due to their dependence on different assumptions on the characteristics of the tremor, which normally shows a high degree of inter- and intra-subject variability. As the devised solution for PHT elimination is based on data-driven methods, and any possible foreseeable signal for voluntary movements can be generated, we believe that employing pseudo-synthesized signals adequately addresses this challenge and eliminates the need for making unrealistic assumptions on tremor behavior. To quantitatively measure the performance of PHTNet, the MSE criterion was used:

$$\mathrm{MSE} = \frac{1}{T}\sum_{t=1}^{T}\left(m_v^{(\mathrm{GT})}(n) - \hat{y}_v(n)\right)^2, \tag{4.29}$$

where $m_v^{(\mathrm{GT})}(n)$ is the ground truth for the voluntary component, $\hat{y}_v(n)$ is the estimated voluntary component, and T is the length of the input sequence to the network. We have employed the MSE criteria in various evaluation scenarios to fully investigate the performance of PHTNet.

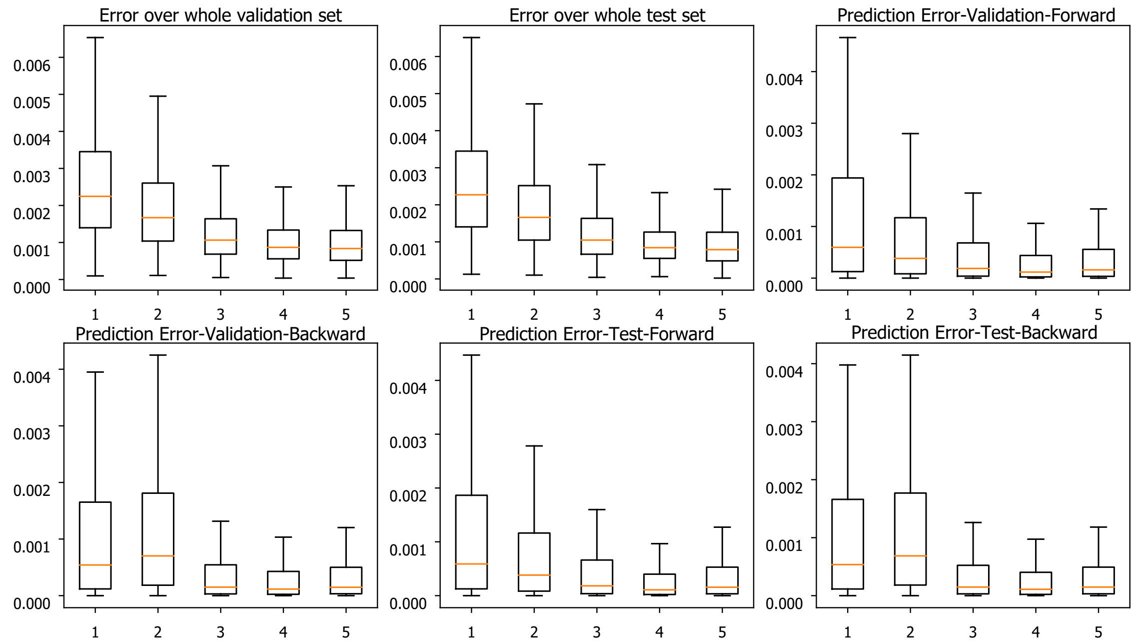


Figure 4.12: Comparison of Normalized Mean Squared Error (y-axis) over validation and test sets for different lengths of input sequence in seconds (x-axis). Performance of the PHTNet in each test case is shown in boxplots, where the orange line indicates the median, the box indicates the range between 25% to 75% quartiles, and the lines indicate the standard deviation range. "Prediction Error" indicates the error for only the last sample of estimated voluntary signal. "Forward" and "Backward" indicate which path of the PHTNet is used.

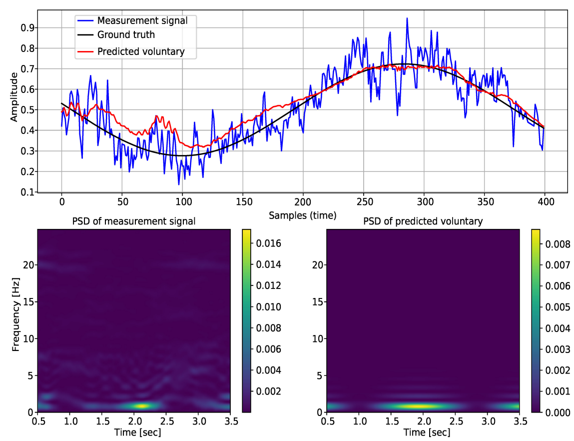
Moreover, we have employed a qualitative approach, to visually investigate the performance of PHTNet in PHT elimination and demonstrate its superior estimation accuracy compared to its counterparts. In visual inspection, we can verify if the output of PHTNet follows the low frequency trend of the measurement signal, i.e., the voluntary component, that we intuitively expect. The visual investigation is performed to check if the designed neural network is operating in the expected way. It should be noted that visual inspection is not a means of evaluating the performance of PHTNet and only serves as the preliminary verification of the network.

4.3.5 Simulation Results

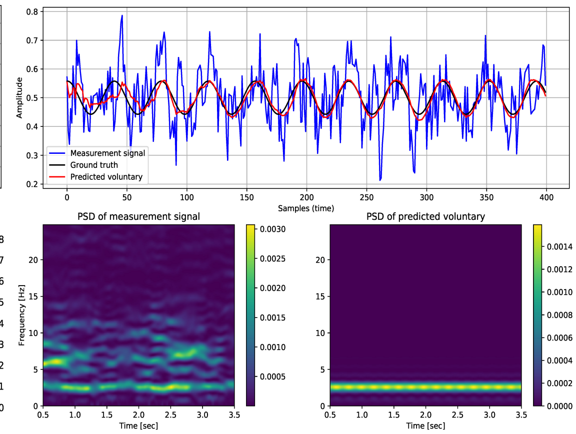
In this section, the performance of PHTNet is evaluated in several scenarios. Also, supporting results on the effectiveness and capacity of the devised PHTNet for PHT extraction problem are also discussed.

Quantitative Evaluation

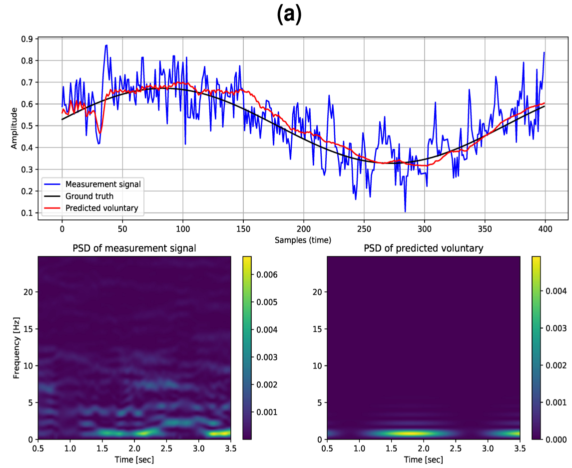
Here, the results of a quantitative evaluation over the pseudo-synthesized validation and test sets are reported. As stated previously, the benefit of employing pseudo-synthesized validation and test sets is the possibility of numerical performance evaluation. Static tremor



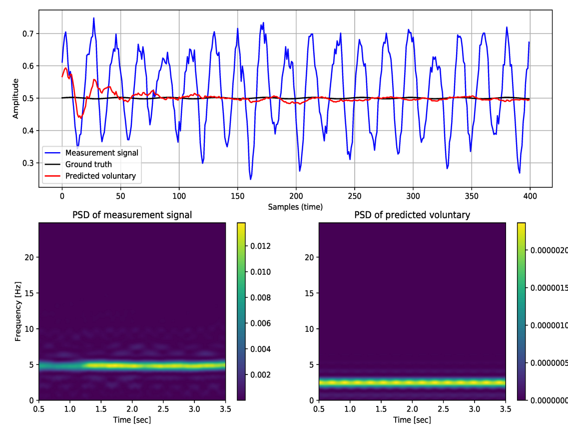
(a)



(b)



(c)



(d)

Figure 4.13: Visualization of the network's output when the pseudo-synthesized evaluation signals from validation and test sets are fed to the network. Please note that due to the pseudo-synthesized nature of these signals, the known synthetic ground truth is mixed with pathological data to augment the input space and enforce the model to learn how to extract the pathological tremorous motion. The PSD of input and output signals are also included to demonstrate the transformation and manipulation of the spectral contents. The PSD plots are obtained by sliding a FFT window of size 50 samples and overlap size of 45 samples over the signals. The color bars also determine the density of each frequency at each time and its dimension is $\frac{Amplitude^2}{Hz}$. **(a),(b)** Two instances of the synthesized evaluation signals from the patients in the validation set. **(c),(d)** Two instances of the evaluation signals generated from the static tremor recordings of patients in the test set. Please also note that the presented PSD diagrams are only obtained for visualization purposes and their properties do not influence any parameter or hyper-parameter of the PHTNet.

recordings of 16 subjects for validation and 16 subjects for testing are combined with voluntary components to synthesize data for quantitative evaluation. Please note that the tremor assessments of one patient only belong to the training set, or the validation set, or the test set to avoid leakage of data from training set to the other two sets. The network's output for different pseudo-synthesized inputs is shown in Figure 4.13. Please note that the PSD diagrams in Figure 4.13 are derived by sliding a Fast Fourier Transform (FFT) window of

Table 4.5: Results of a quantitative evaluation of the network in different testing scenarios.

	Validation set	Test set
MSE of estimation for a complete segment	0.00111	0.00104
MSE of estimation for the last sample of segment over 24300 trials (*forward cells*)	0.00056	0.00049
MSE of estimation for the first sample of segment over 24300 trials (*backward cells*)	0.00052	0.00048

size 50 samples and overlap size of 45 samples over the signals. A high overlap size is selected to produce a smooth representation of the spectral contents of the signals. It is worth mentioning that the aforementioned properties are only employed for visualization purposes and do not reflect any parameter or hyper-parameter for the PHTNet. It is worth mentioning that to further investigate the performance of PHTNet and to discover the way it manipulates the spectral contents of the input signal, the PSD of the input and output pairs are also shown. The numerical results of the quantitative tests are presented in Table 4.5. As the main goal of this work is to develop a PHT estimation framework for online and offline applications, we believe that it is necessary to monitor the estimation error for the forward and backward cells, separately. In addition, as the last samples of the output sequence in the forward direction are mainly employed for online applications, and the initial samples of the output in the backward direction are employed in offline applications, we have also measured the estimation error for the last sample in the forward cells and the estimation error associated with the first sample in the backward cells.

Although the examples presented in Figure 4.13 demonstrate how the spectral contents of the involuntary component are manipulated, a more rigorous analysis is required to take the whole validation and test sets into account and statistically investigate the effectiveness of the PHTNet framework for PHT elimination. To statistically investigate performance of the PHTNet in comparison to the state-of-the-art methods on PHT removal, and due to the fact that we need to have pseudo-synthesized action tremor data to accurately measure the error in estimation, PHTNet along with three well-regarded techniques on PHT removal, i.e., BMFLC, EBMFLC, and WAKE, were examined over the validation and test sets of pseudo-synthesized action tremor. To statistically compare the four methods, the "Analysis of Variance (ANOVA)" test, and pairwise "Z-test" between each two groups are employed, where the results are shown in Figure 4.14 and Table 4.6. As it is understood, PHTNet shows significantly better performance compared to state-of-the-art methods in the literature. It is worth noting that in the boxplots in Figure 4.14a, the red line indicates the mean value, the box indicates the range between 25% to 75% quartiles, the black lines indicate the standard deviation range, and red crosses indicate the outliers. Furthermore, Figures 4.14b and 4.14c show multiple comparison of the mean performance of the four methods over the validation and test sets. The dashed lines indicate the upper and lower 95% confidence bounds for the error of each method. The disjoint confidence intervals of PHTNet with other methods indicates that the mean performance of the PHTNet is significantly different from other methods. It is worth clarifying that the plots in Figures 4.14b and 4.14c show the mean Normalized MSE (shown as a circle) and the 95% confidence intervals (shown as dashed lines) for each of the four techniques that are evaluated over the validation and test sets. The y-axis of the plots represents the label of each technique and the disjoint area between the dashed lines of each two methods verifies that each two methods have significantly different mean performance from each other.

Furthermore, to statistically investigate the efficacy of PHTNet in manipulating the spectral contents of hand motion signals, the FFT of the input and output signals in the validation and test sets are derived and statistically compared. Due to the large number of samples in each of the validation and test sets ($24,300$ signal pairs for each set), we employed "Z-test" for statistical comparison of the input and output groups. We performed D'Agostino and Pearson's test [249] to check if the data samples for each frequency and for each of the input and output groups follow a normal distribution. After verifying the normality of data samples, we employed the "Z-test" to extract the confidence intervals by setting $Z = 1.96$, which corresponds to 95% confidence bounds. In fact, the confidence interval states that the probability that the mean of data samples occurs in the range of $(-1.96 * \sigma/\sqrt{n},\ 1.96 * \sigma/\sqrt{n})$ is equal to 0.95, where σ and n represent the standard deviation and the sample size, respectively. As shown in Figures 4.15a and 4.15b, the mean and confidence intervals for the input and output pairs in the validation and test sets are completely distinct and the extracted $p-value$ between the input and output pairs, which is $\ll 0.01$ at each frequency, verifies that applying the proposed PHTNet was resulted in a significant change in the spectral information of the signal by removing the tremor, such as the illustrative examples shown in Figure 4.13. In addition, Figures 4.15c and 4.15d show the mean FFT along with its standard deviation intervals (mean $\pm$ standard deviation) for the input-output pairs. Please note that all of the presented figures do not include the spectral content corresponding to 0 Hz to better scale the figures and present a more detailed picture. It is worth noting that the results shown in Figures 4.15c and 4.15d represent the variability of the spectral contents for each group around the mean spectral value of the population. Although the efficacy of the PHTNet in damping the spectral contents of the PHT is reflected in Figures 4.15c and 4.15d, it is not readily inferred whether the spectral contents of the input and output populations are significantly different or not. Thus, the plots in Figures 4.15a and 4.15b are generated to reveal the p-value between the two populations at each frequency point and also to illustrate the position of the two spectral populations with respect to each other. As demonstrated in Figure 4.15, a significant decrease of power in the spectral content above 3 Hz is observed, which reveals the filtering behavior of PHTNet, although we did not explicitly design a spectral filter. More importantly, unlike conventional spectral filtering methods which normally result in phase-lag (delay) in their output signals, the proposed PHTNet advances the input signal by predicting the voluntary component in the next time.

Predictive Behavior Analysis

As mentioned earlier, one goal of this work is to predict the voluntary motion of an individual in a one-step-ahead-of-time fashion, which is of significant importance in robotic rehabilitation technologies. To address this goal, we devised a novel training strategy to equip PHTNet with predictive functionalities. Table 4.5 shows the numerical evaluation of the predictive behavior of PHTNet over 48600 input-output pairs for both validation and test sets. However, to visually inspect the output signal and verify if it actually advances

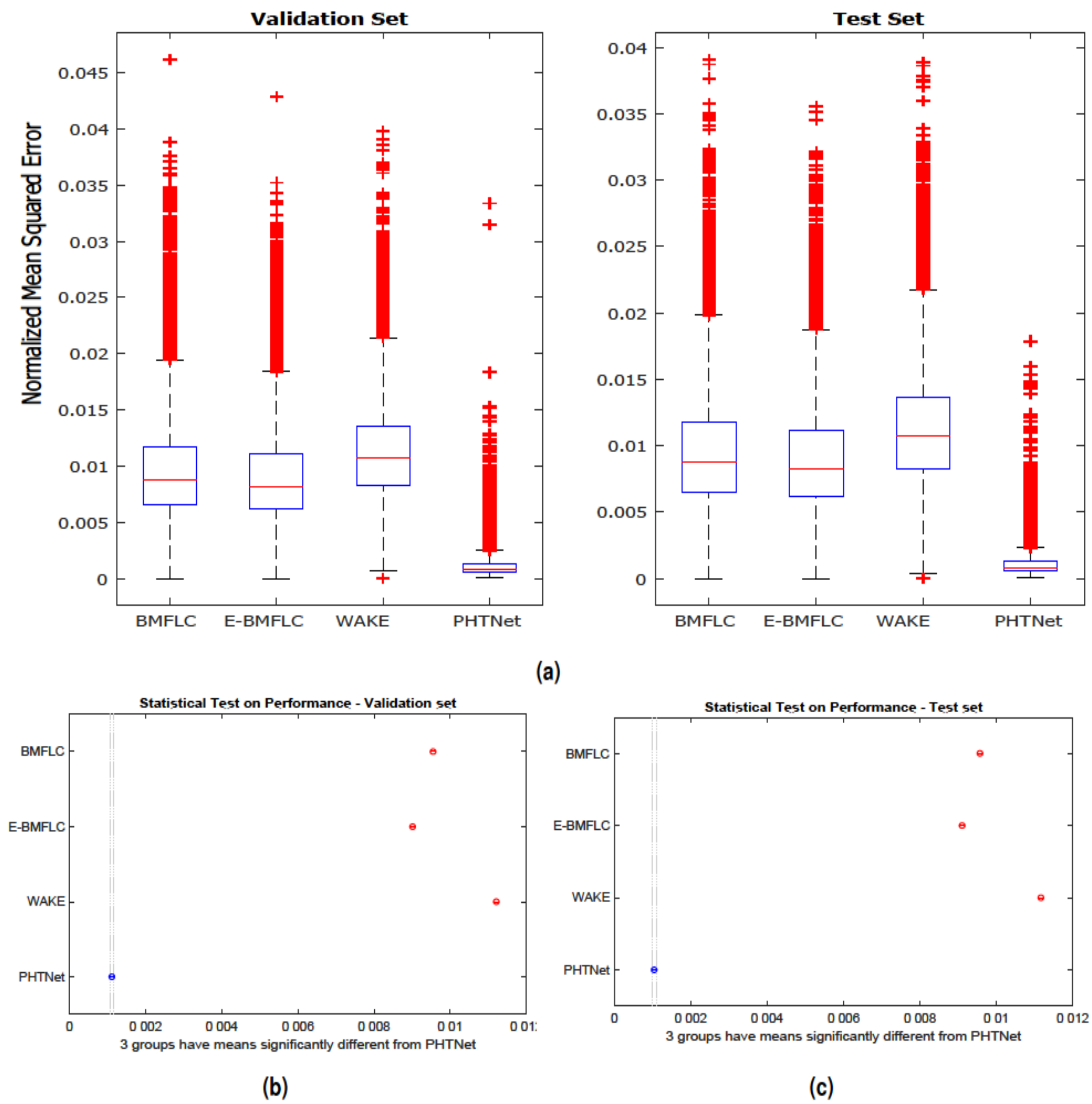


Figure 4.14: **(a)** Comparison of the performance between BMFLC, EBMFLC, WAKE, and PHT-Net on the validation and test sets. The red line in the boxplots indicates the median performance and the box indicates 25% and 75% quartiles. The dashed lines show standard deviation and the red crosses show outliers. **(b)** and **(c)** Multiple comparison of the mean performance of the four methods over the validation and test sets. The dashed lines indicate the upper and lower 95% confidence bounds for the estimation error of each method and the y-axis represents the labels of compared PHT removal techniques.

the input signal, in this part, we employ pure sinusoidal signals as inputs to PHTNet. The benefit of feeding the network with sinusoidal signals is that we can clearly observe the status of the output signal with respect to the input, without any interference from the involuntary component of the hand motion signals. Figure 4.16 shows the capability of the trained network in predicting the voluntary component.

Table 4.6: Comparison of the performance between PHTNet and three well-regarded PHT processing frameworks. The $p-values$ are derived based on 95% confidence intervals. The numbers in the “Improvement” rows represent the improvement in the mean Normalized MSE obtained when moving from other techniques to the PHTNet.

Dataset		BMFLC-PHTNet	EBMFLC-PHTNet	WAKE-PHTNet
Validation	p-value	$5.96E-08$	$5.96E-08$	$5.96E-08$
	Improvement	+88.42%	+87.78%	+90.18%
Test	p-value	$5.96E-08$	$5.96E-08$	$5.96E-08$
	Improvement	+89.58%	+89.01%	+91.07%

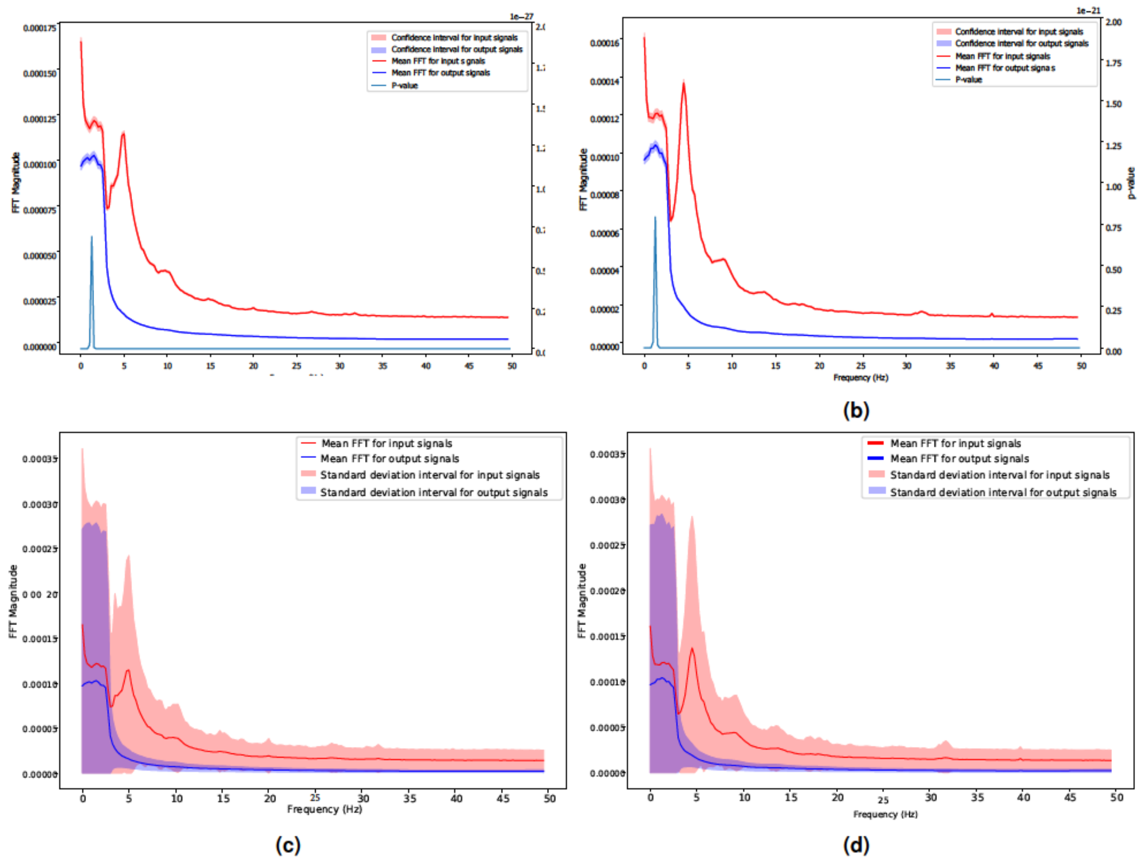


Figure 4.15: **(a)** and **(b)** Visualization of the 95% confidence boundaries for the spectral contents of input-output pairs in validation and test sets, respectively. The solid lines indicate the mean of spectral contents and the highlighted area indicate confidence boundaries. Also note that the y-axis on the right represents the $p-value$ between the spectral contents of the input and output signals. **(c)** and **(d)** Visualization of standard deviation boundaries for input-output pairs along with the mean of spectral contents of input and output signals in validation and test sets, respectively. Please note that the spectral contents of the input signals are shown in “red” color, while those of the output signals are depicted with “blue” color. The dominant activity at around 5 Hz is only observed in the input signals (“red” color), which is expected due to the strong power of pathological tremor around 5 Hz.

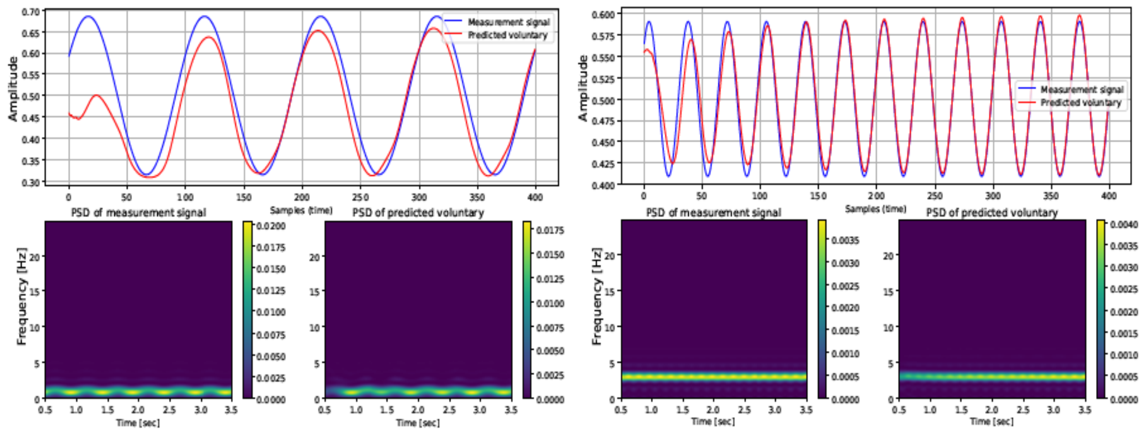


Figure 4.16: Visualization of the predictive capability of the network over special test signals.

Qualitative Performance Monitoring

Qualitative inspection of the PHTNet framework was performed on the action tremor recordings due to the absence of a valid framework to extract the voluntary component of hand motion signals. As the performance of PHTNet over real action tremor signals cannot be numerically reported, performance monitoring is conducted through visual inspection, where one checks if the estimated voluntary component is aligned with the expected low-frequency trend in the signal or not. Figure 4.17 shows 12 instances of the processed signals which include action tremor recordings from the training, validation, and test sets. To further investigate the performance of the PHTNet framework, we compared our results with BMFLC [126], EBMFLC [81], and WAKE [41]. While the FLC-based methods, e.g., BMFLC and EBMFLC, are focused on modeling the spectral contents of the measurement signal with a linear combination of spectral components, the WAKE method employs spectrotemporal techniques and Kalman filtering to decompose the measurement signal into the two components of motion. Moreover, to visually inspect the performance of the forward (predictive) path of the PHTNet over action tremor data, the action tremor signals shown in Figure 4.17 are also processed by the forward path and the results are shown in Figure 4.18. As it is shown, the output of PHTNet perfectly follows the voluntary component of motion that we visually expect.

PHTNet over Healthy Controls

To investigate the performance of the PHTNet over hand motion recordings from healthy individuals, 2 set of signals from 2 healthy volunteers were recorded. Two healthy male individuals aged 28 and 37 participated in the data collection procedure and the acceleration of their hand motion is recorded with Trigno Avanti Wireless System PM-W05 (Delsys Inc.). Trigno Avanti sensors have a built-in 9 DOF inertial measurement unit which can relay acceleration, rotation and earth magnetic field information. The sensitivity of the sensor is set to $\mp 16g$ and the sampling rate by default is 370.37 Hz. However, for this experiment the recordings are downsampled to 100 Hz to match the training data of the PHTNet and the results are shown in Figure 4.19. Performing this test is crucially important to check the performance of the PHTNet over hand motion recordings without tremor component. As it is observed in the figure and was expected before implementing the test, the estimated signals by PHTNet perfectly match the input signals, as no manipulation should be applied on signals without tremor component. It should be noted that the kinematic data for the hand from healthy individuals was recorded with a different apparatus from that collected in the original dataset. While this was due to an unforeseen situation whereby the original equipment was not available, we believe that the use of a different but clinically accepted device to assess the functionality of the PHTNet provides a good opportunity to examine the generalization of the network and its level of independence of the recording device. In fact, the results shown in Figure 4.19 not only show the flawless performance of the PHTNet in processing the kinematics of hand but also reflect a high degree of generalization over the characteristics of PHT and the independence of the PHTNet with regard to the recording device.

4.3.6 Discussion

From the results of the experiments over the validation and test sets presented in Figures 4.13 and 4.17, and Table 4.5, we can clearly observe the superior performance of PHTNet in accurate estimation of the voluntary hand motion from pseudo-synthesized and real action tremor signals. The examples presented in Figure 4.13 include different possible cases for voluntary component, i.e., [high vibration amplitude - low frequency] in Figures 4.13a and 4.13c; [low vibration amplitude - high frequency] in Figure 4.13b, and [very low (near zero) vibration amplitude] in Figure 4.13d. Observing the PSDs of input-output pairs also clearly shows how the high frequency components are damped, while at the same time, the low frequency trend in the measurement signal is magnified. RNNs typically yield inaccurate outputs for the first few samples of the input sequence and as more information is fed into the network, the estimation process becomes more accurate. This natural behavior of RNNs is also observed in the validation examples shown in Figure 4.13, which confirms the necessity to employ a bidirectional architecture with two separate processing pipelines within the PHTNet. Thus, the bidirectional architecture empowered the PHTNet to maintain the required level of accuracy for PHT elimination tasks, when it is used for online applications through forward cells, and for offline applications via its backward cells. It is important to note that the instances presented in Figure 4.13 are derived by employing the forward cells of PHTNet. To evaluate the network in online and offline applications, the estimation accuracy is shown in Table 4.5 for the next time sample when forward cells and backward cells are employed. The results clearly show the accuracy of PHTNet in estimating the voluntary component, when enough information is fed to the network.

In addition to the instances shown, a rigorous statistical analysis is performed on all of the samples in the validation and test sets to fully examine the operation of PHTNet. In this regard, standard deviation and 95% confidence boundaries were calculated over $24,300$ samples in the validation and test sets, as shown in Figure 4.15. These results suggest that PHTNet operates as a low-pass spectral filter; however, the predictive behavior of PHTNet makes it distinct from any previously known spectral filter. In Figure 4.15, a dominant activity at around 5 Hz is observed in the spectral domain of input signals ("red" color), which verifies the reported characteristics of the PHT in the literature, i.e., the PHT in PD and ET patients commonly occurs in $4 - 6$ Hz and $4 - 8$ Hz [98, 135], respectively, which results in accumulation of power in frequencies around 5 Hz. In the qualitative performance monitoring part, the output of PHTNet is compared with well-regarded, recent works in the field of PHT estimation, i.e., BMFLC, EBMFLC, and WAKE. As shown in Figure 4.17, our proposed method provides a smooth and tremor-free output, which is compatible with the visual trend that we expected, and is also robust to sudden high amplitude vibrations. Please note that for the instances presented in Figure 4.17, the backward cells of PHTNet are employed to extract the voluntary component of hand motion signals. In this setup, the PHTNet slides over the action tremor signal and the estimated voluntary component is obtained. To only keep the accurate part of the output signal, which in this case consists of the first few samples of the estimated signal, PHTNet advances for 50 samples and again the estimation is performed. This process continues until the whole sequence of action tremors is processed. In this context, the network slides over the measurement signal and outputs the voluntary component. To show the generalization of the network, we have included action tremor instances from training, validation, and test sets. To further assess the predictive behavior of the network in addition to the numerical results reported in Table 4.5, which clearly illustrate the accuracy of estimation when forward cells are employed, we fed the network with pure sinusoidal signals to investigate its response to the inputs and verify if the network shows any predictive behavior.

This paper proposes the design and implementation of a novel voluntary motion prediction and tremor removal technique that can be used for enhancing assistive devices and clinical settings. Although the proposed trained technique significantly performed better than all existing approaches, it requires relatively stronger computational support to be implemented due to the deep neural structure. It should be noted that the training of the model is completed in this paper, and the model can be used as a ready, plug-and-play trained algorithm without the need for retraining. However, utilization of any deep neural network with memory gates, requires sufficient computational power, which can be a limiting factor if the computational resources are strictly limited. Thanks to the power of new processing technologies, this challenge will not be very concerning but should be considered when implementing. In addition, we are working on a cloud computing approach for this work, which can be used for minimizing the need for having on-site computational power. Also, in order to enhance this aspect of the technique, we have an ongoing research to optimize the design of the PHTNet and implement hybrid and shallower models with comparable performance. In addition to the above points, it should be also highlighted that the accurate predictability of the proposed technique is a novel and unique feature, which do not exist in conventional techniques. However, the achieved horizon of prediction was limited. We are planning to augment the input space with other biological modalities and biomechanical models while improving the prediction ability of the technique to further enhance the horizon. Lastly, another future direction for this work would be to expand the size of the dataset employed and investigate the performance of the framework when the kinematics of motion in all dimensions are jointly fed to the network to potentially enhance the perception of the network over the voluntary action and increase its predictive horizon.

Table 4.7: Different training and testing schemes based on the employed two datasets.

Scheme	Training	Validation	Testing	Summary
Motus Dataset	4 subjects	1 subject	5 subjects	The leave-one-out is performed for 5 subjects and 5 different networks are trained. The 5 subjects for training/validation contain only resting tremor data and the 5 subjects for testing contain only action tremor data. Quantitative and qualitative tests are feasible on this dataset.
Smartphone Dataset	9 subjects	1 subject	0 subject	This dataset contains only resting tremor recordings of 10 subjects and thus leave-one-out strategy is performed for 10 different times. Quantitative testing is performed on this set but for the qualitative testing, the action tremor data from Motus Dataset is used.

4.4 The HMFP-DBRNN Framework

In this section, the previously introduced RNN-based architecture for PHTNet is employed over two smaller datasets, which are publicly available for benchmarking purposes. In addition, the possibility of merging the two datasets in order to populate larger dataset is examined and the results are discussed. Since the methodological details are already discussed in Sections 4.3.2 and 4.3.3, the next section will elaborate on the simulation results.

4.4.1 Simulation Results

To train and evaluate the HMFP-DBRNN framework, two real datasets, namely “Motus dataset" and the “Smartphone dataset" were employed, which are already described in Sections 4.2.4 and 4.2.4. As shown in Table 4.7, to train the proposed network, two different schemes are employed to show the performance and robustness of the learning procedure. For all the cases, the corresponding rest tremor signals are collected and all the possible 4 seconds-length segments with 1 second overlap are extracted and then segments are randomly combined with synthesized voluntary component. To produce a validation set for each scheme, the “leave-one-subject-out” approach is employed and training of the network is repeated for the number of subjects in the two schemes. In each scheme and for each round, a training epoch of 2000 trials and a validation epoch of 1000 trials are generated and employed in the training and evaluation. The numerical results of the implementations according to Table 4.7 are presented in Table 4.8.

To evaluate the performance of the framework, therefore, two approaches are taken, which are “Quantitative” and “Qualitative” methods. The quantitative method relies on the validation set for which the ground truth associated with the voluntary component is available and precise measurement of the performance is plausible. Therefore, the quantitative performance is measured in each case and for each round of training and then is averaged over all the rounds. On the other hand, the qualitative method operates on real action tremor recordings and as the ground truth of the voluntary component in the action tremor signal is not available, the evaluation is only based on visual inspection. In this case the output of

Table 4.8: Quantitative testing of the network over the two different schemes.

Scheme	MSE	NRMSE	PRF
Motus	0.001	0.0632	0.004
Smartphone	0.0019	0.0872	0.0076

the framework is visually compared with the output of other competing frameworks in the literature and the inferences are made based on the plots.

In the training process, the loss function is defined as: Loss (MSE) $= \sqrt{\sum(\boldsymbol{c}^{(\mathscr{V})}\boldsymbol{f} - \hat{\boldsymbol{c}}^{(\mathscr{V})})^2}$, and the Adam optimizer [248] with the learning rate of 0.0001 is deployed to solve the loss minimization problem. To further investigate the performance of the trained network over the validation sets, the following error measures are also deployed

$$\text{Normalized Root MSE} = \frac{\sqrt{MSE}}{max(\boldsymbol{c}^{(\mathscr{V})}\boldsymbol{f}) - min(\boldsymbol{c}^{(\mathscr{V})}\boldsymbol{f})}, \tag{4.30}$$

$$\text{Power Ratio Factor (PRF)} = \frac{MSE}{\frac{1}{400}\sum(\boldsymbol{c}^{(\mathscr{V})}\boldsymbol{f})^2}. \tag{4.31}$$

Table 4.8 represents the quantitative performance of the proposed HMFP-DBRNN framework in different scenarios.

Figure 4.20 demonstrates two instances of the synthesized test data and the network output. By having a closer look at Figure 4.20, it can be observed that the output of the network, which is the estimated signal for voluntary movement, is one sample ahead of the ground truth, which means that we have successfully fulfilled our vision in designing the network. For qualitative assessment, Figures 4.21(a) and 4.21(b) depict the output of the HMFP-DBRNN framework compared with well-regarded techniques such as BMFLC [126], E-BMFLC [81], and WAKE [41]. As it can be observed, the output of the HMFP-DBRNN is smoother and more robust to the high-frequency variations, which are tremor components. In addition, it is worth mentioning that the output of the HMFP-DBRNN frameworks is generated in a predictive fashion, while the BMFLC and E-BMFLC methods operate in an offline fashion.

4.4.2 Data Fusion for Training the RNN

Due to the data-hungry nature of deep neural networks, and unavailability of large datasets in medical fields, the application of deep learning methods may seem to be still limited. Typically, the networks trained over shallow datasets do not generalize well, and there could be the possibility of over-fitting in the model. To overcome this challenge, in this section, we take the HMFP-DBRNN framework one step forward and train it over two different hand motion datasets which are recorded by accelerometer and gyroscope sensors. In other words, the paper proposes a data fusion strategy and investigates the feasibility of combining the two different multimodal datasets, collected under two different conditions with two different experimental setups, in order to train a tremor extraction neural network. This improves the generalizability of the model.

Table 4.9: Different training and testing schemes based on the employed two datasets. Scheme 1 is Motus Dataset, Scheme 2 is Smartphone dataset, and Scheme 3 is the combination of the two datasets.

Scheme	Train	Validate	Test	Summary
1	4	1	5	The 5 sets of rest tremor are used for training and validation, and 5 sets of action tremor are used for qualitative evaluation. The leave-one-out is performed on 5 subjects for 5 different substitution. Quantitative and qualitative tests are feasible on this dataset.
2	9	1	0	10 sets of rest tremor are employed for training and validation, as well as qualitative evaluation. Quantitative evaluation is not feasible on this dataset.
3	12	3	5	In the training phase, we used the recordings of 12 subjects (4 Motus + 8 Smartphone) and 3 subjects (1 Motus + 2 Smartphone) are left for the validation step. Quantitative and Qualitative tests are feasible.

Data Fusion Strategy

The rational behind the fusion idea is the fact that neural networks have shown superior performance in so many cases, however, require relatively large datasets to surpass the performance of classical methods. By fusing a number of small datasets, we can potentially train larger neural networks and improve the performance of tremor estimation frameworks.

The main characteristic of tremor that makes it different from the voluntary motion is its spectral range, which typically, appears as high-frequency contents. Therefore, the important property of the input, fed to a network for training phase, is having a combination of high-frequency components representing tremor, and low-frequency components representing the voluntary movement. As stated in Reference [130] and by comparing the PSD of the rest tremor signals recorded by accelerometer (Smartphone Dataset) with gyroscope (Motus Dataset), as shown in Figure 4.22, it can be observed that these two types of measurements illustrate very similar patterns for the spectral components related to the tremor. The main difference observed in the two groups is with regards to the extremely low frequency components. The reason behind the low frequency variations is associated with the data recording process. More specifically, in the Smartphone Dataset, the hand of the patient is kept steady and the hand motion is recorded only when the hand is steady. On the other hand, in the Motus Dataset, the signals contain the transitions to the steady state and thus very low-frequency contents are visible in the PSD. Thus, to estimate and remove the tremor out of the measurement signals and output the voluntary component, the aforementioned two datasets can be used jointly, as they share similar spectral characteristics.

As it is shown in Table 4.9, three different schemes under which the HMFP-DBRNN is trained and validated are described. For all the three schemes, segments of 4 seconds length are extracted and then randomly combined with synthesized voluntary components to train the network. For the validation phase, the mean of performances in the “leave-one-subject-out” strategy is considered. For the “Motus+Smartphone” case that we have the recordings of 15 individuals in total, in each round, recordings of 8 subjects from Smartphone dataset and 4 subjects from Motus dataset are used for training and the remaining 3 subjects are

Table 4.10: Quantitative testing of HMFP-DBRNN.

Scheme	MSE	NRMSE	PRF
Motus	0.001	0.0632	0.004
Smartphone	0.0019	0.0872	0.0076
Motus+Smartphone	0.0022	0.0938	0.0088

participated in the validation set and this process is repeated for 10 times. 2000 training examples and 1000 validation samples are generated for each round of training.

Simulation Results

Likewise the evaluation procedure presented in Section 4.4.1, "Quantitative" and "Qualitative" methods are employed to evaluate the estimation accuracy of the HMFP-DBRNN. The quantitative method is used to numerically measure and report the performance of the network over the validation set. MSE, NRMSE, and PRF are the employed performance evaluation metrics. Moreover, to optimize the network, the Adam optimizer [248] with the learning rate of 0.0001 is deployed. Table 4.10 represents the quantitative performance of the proposed HMFP-DBRNN framework in different scenarios.

Due to absence of a robust method to extract the voluntary component of hand motion signals, we can not numerically evaluate the network's output for action tremor signals, and as such, we rely on visual inspection for qualitative evaluation of the HMFP-DBRNN. To this aim, the HMFP-DBRNN, BMFLC [126], E-BMFLC [81], and WAKE [41] methods are compared with each other over same action tremor signals and the results are presented in Figure 4.23. As it is clearly observed, the output of HMFP-DBRNN in "Motus + Smartphone" scheme is almost identical to its counterparts in other schemes and represents smoother signal compared to the output of other methods in the literature. The numerical and plotted results suggest that fusing datasets of the same nature not only provides rather similar results but also helps the network to generalize better and cover a wider variety of input signals.

In this work, we investigated the idea of training a neural network by fusing multimodal datasets, which have recorded the same phenomenon, i.e., PHT, but with different devices. The multimodal fusion of datasets is evaluated based on our recently proposed HMFP-DBRNN framework, which offers the state-of-the-art results in the field of tremor extraction. As the results of quantitative and qualitative evaluations suggest, fusing the datasets, under certain conditions and at the cost of slightly higher estimation error, helps the network to generalize better over the subject domain.

4.5 Summary

In this chapter, the major contributions of the thesis in PHT processing were presented. As discussed in Section 4.1, the main objective of PHT processing frameworks is to estimate the voluntary and involuntary components of hand motion in the individuals affected by age-related neurological disorders, which share PHT as a common symptom. Precise estimation and extraction of the two components play an imperative role in the quality of services delivered by various types of ADs, including but not limited to robotic rehabilitation devices, tele-robotic surgery technologies, and smart spoons. Towards this goal, a multi-rate and adaptive processing framework was developed based on Kalman filters and wavelet transformations. Referred to as the WAKE, the proposed framework adapts its filtering parameters in real-time to better track the dynamical changes in the characteristics of PHT. Then, a data-driven solution based on RNNs was proposed, which we referred to as the PHTNet. The proposed PHTNet framework addresses a major bottleneck in the field of PHT analysis, i.e., the unavailability of ground truth for different underlying components of hand motion. Furthermore, the PHTNet provides enhanced estimation accuracy for discriminating the voluntary and involuntary components of hand motion. The RNN-based solution was examined over a large and comprehensive dataset of hand motions in patients with PD and ET and statistically meaningful results were reported. Moreover, the efficacy of the RNN-based solution was investigated over two online available datasets and demonstrated the state-of-the-art accuracy in this domain. The next chapter of the thesis will focus on a data-driven screening protocol for differential diagnosis of PD from ET based on the kinematics of hand in the affected individuals. Besides, another contribution of the thesis will be discussed thoroughly, which presents a novel objective function for training of DL models in deep metric learning scenarios. This objective function enables employment of DL methods in various medical applications where large-scale datasets are not available due to the confidential and sensitive nature of data in this domain.

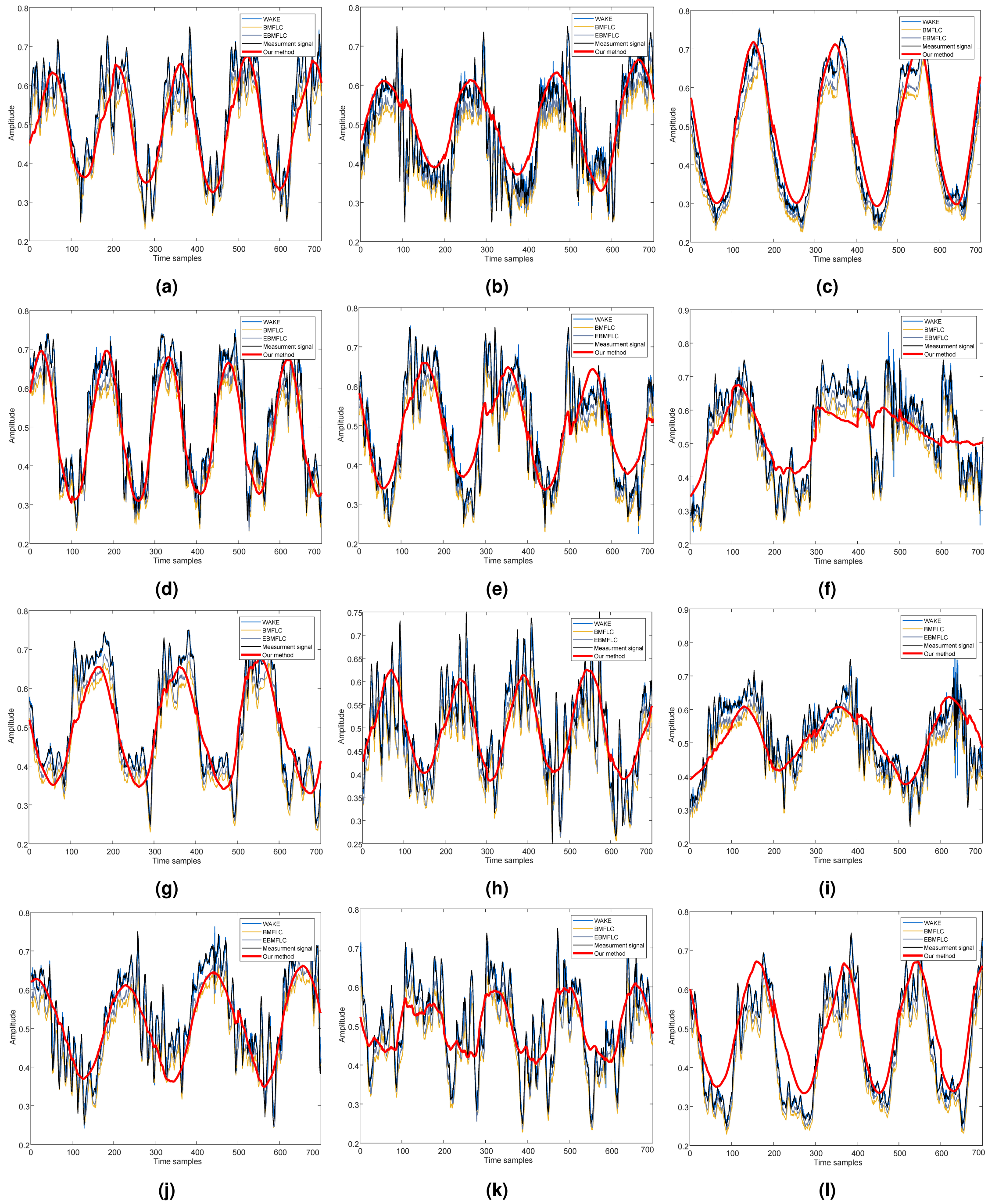


Figure 4.17: Visualization of the network output when real action tremor signals are fed to the network. Our method was compared with three other methods, referred to as BMFLC, EBMFLC, and WAKE. The details of each patient whose signal is shown here are as follows. **(a)** [ET - Right hand - de novo - Training]. **(b)** [PD - Right hand - Under treatment - Validation]. **(c)** [PD - Left hand - de novo - Test]. **(d)** [ET - Right hand - Under treatment - Test]. **(e)** [PD - Right hand - Under treatment - Test]. **(f)** [PD - Right hand - de novo - Test]. **(g)** [PD - Right hand - Under treatment - Test]. **(h)** [ET - Right hand - de novo - Training]. **(i)** [PD - Right hand - Under treatment - Training]. **(j)** [PD - Right hand - de novo - Training]. **(k)** [ET - Left hand - Under treatment - Validation]. **(l)** [PD - Left hand - Under treatment - Validation].

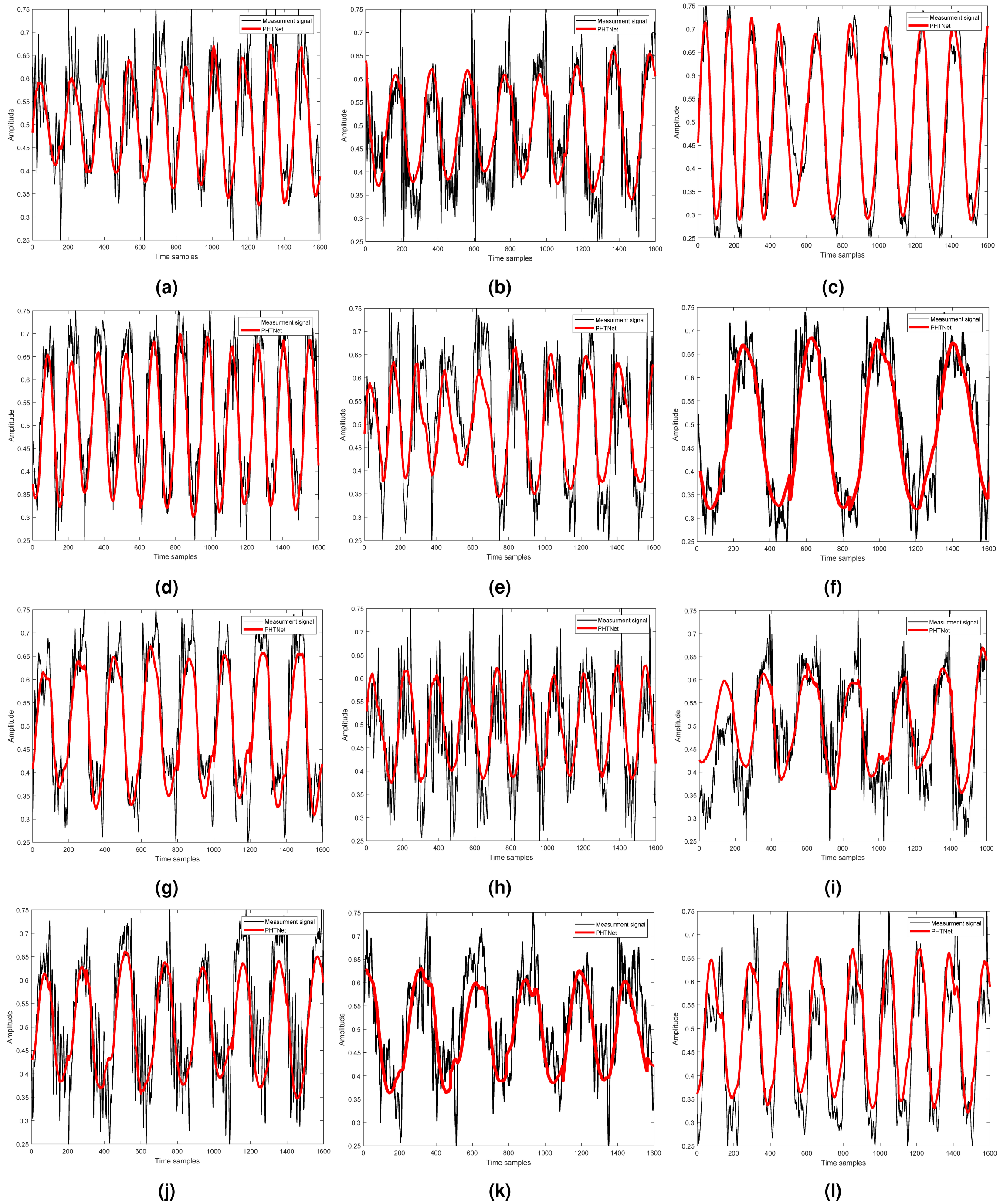
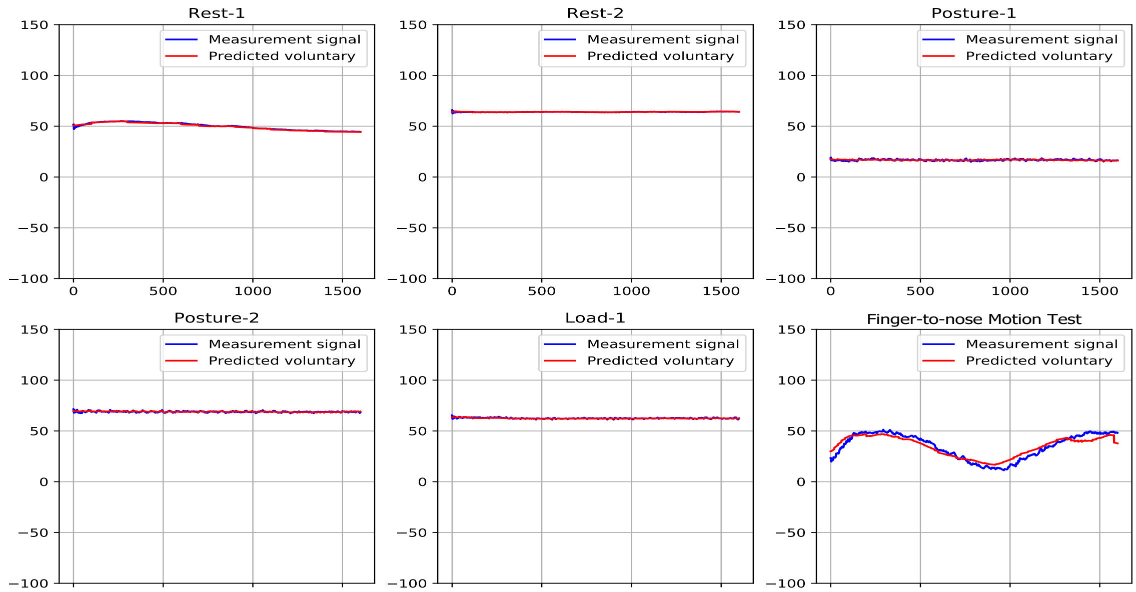
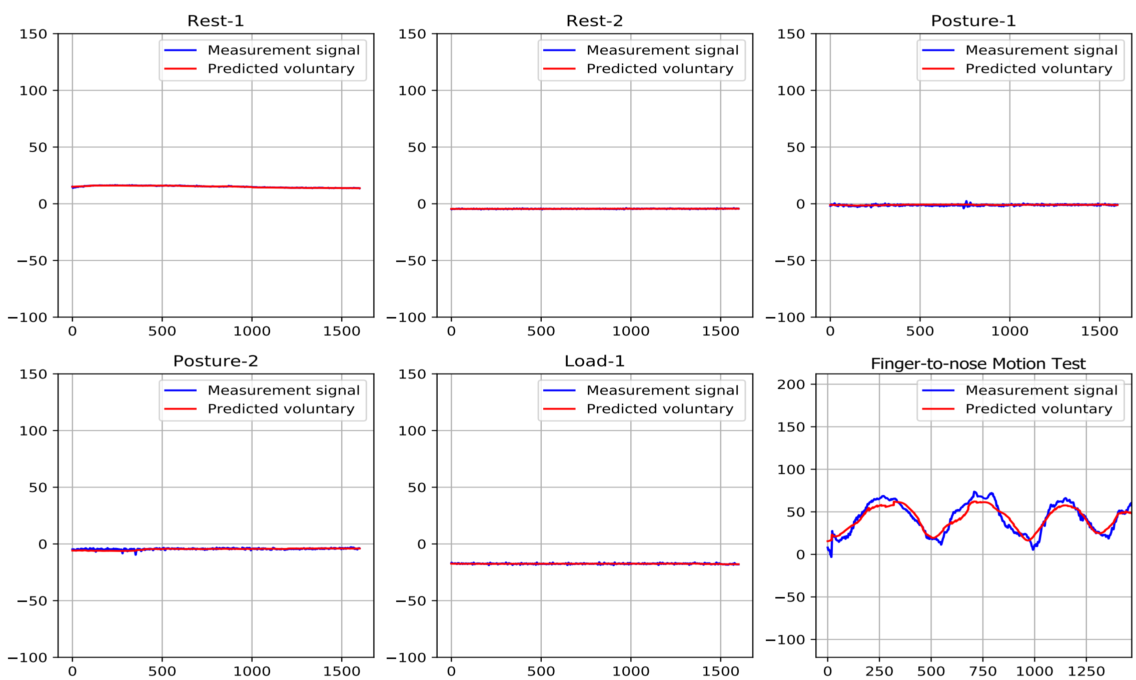


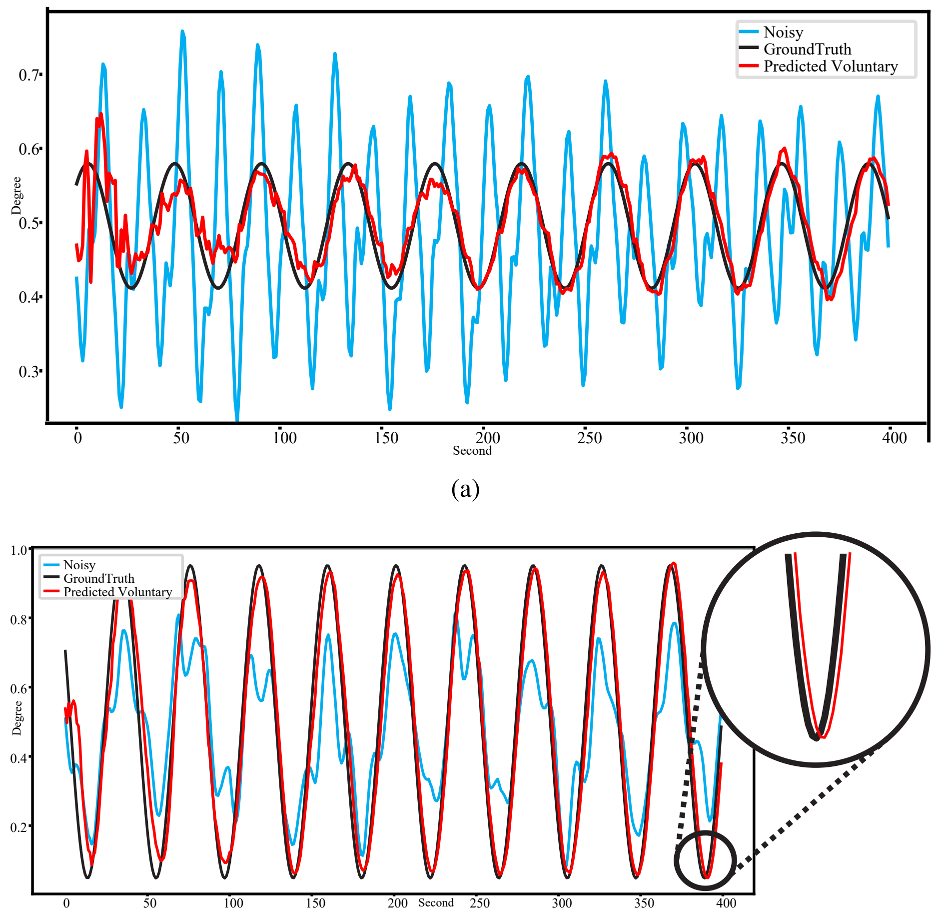
Figure 4.18: Visualization of the network output when real action tremor signals are fed to the network. In this case, the prediction samples are employed. The details of each patient whose signal is shown here are as follows. **(a)** [ET - Right hand - de novo - Training]. **(b)** [PD - Right hand - Under treatment - Validation]. **(c)** [PD - Left hand - de novo - Test]. **(d)** [ET - Right hand - Under treatment - Test]. **(e)** [PD - Right hand - Under treatment - Test]. **(f)** [PD - Right hand - de novo - Test]. **(g)** [PD - Right hand - Under treatment - Test]. **(h)** [ET - Right hand - de novo - Training]. **(i)** [PD - Right hand - Under treatment - Training]. **(j)** [PD - Right hand - de novo - Training]. **(k)** [ET - Left hand - Under treatment - Validation]. **(l)** [PD - Left hand - Under treatment - Validation].





(b)

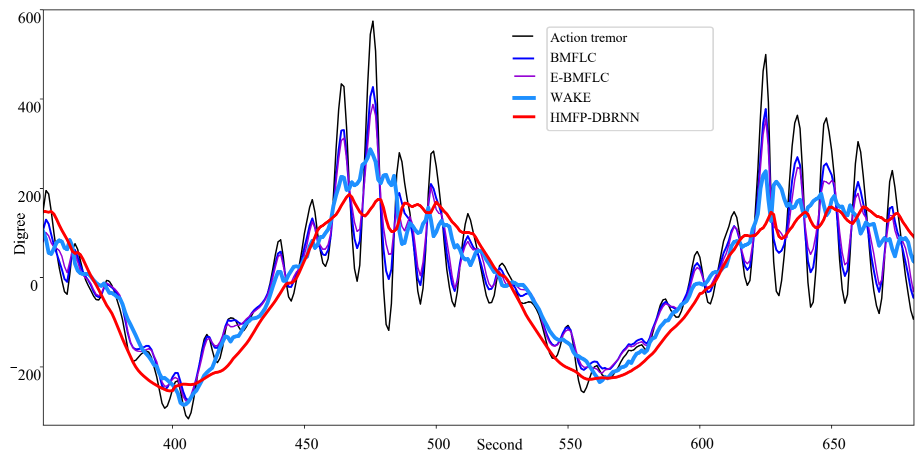
Figure 4.19: **(a)**,**(b)** Results of applying PHTNet on the recordings of static posture tasks and finger-to-nose motion test for the two healthy subjects.

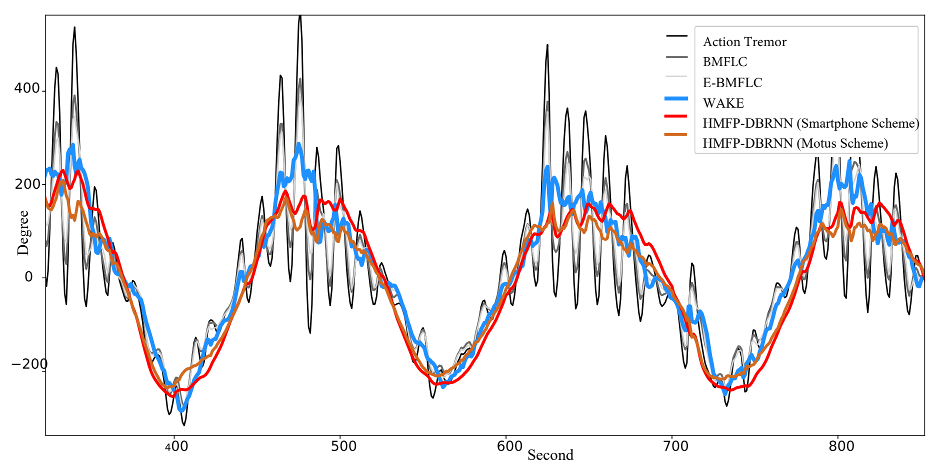


(a)

(b)

Figure 4.20: Visualization of the predicting behavior of HMFP-DBRNN over the synthesized test data. Time samples are on the horizontal axis and the signal amplitude is on the vertical axis.





(b)

Figure 4.21: (a) Performance of the proposed HMFP-DBRNN framework over action tremor recordings in comparison with the state-of-the-art techniques in PHT estimation.

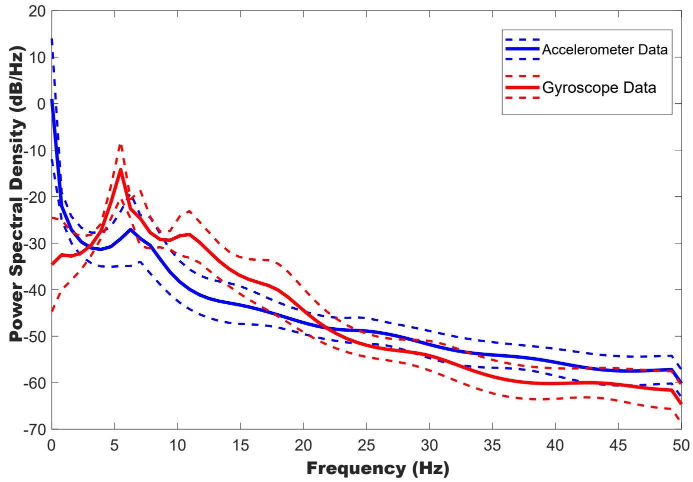


Figure 4.22: Representation of the Power Spectral Density for Motus and Smartphone datasets. The mean of the PSDs along with its standard deviation lines are plotted for the two groups.

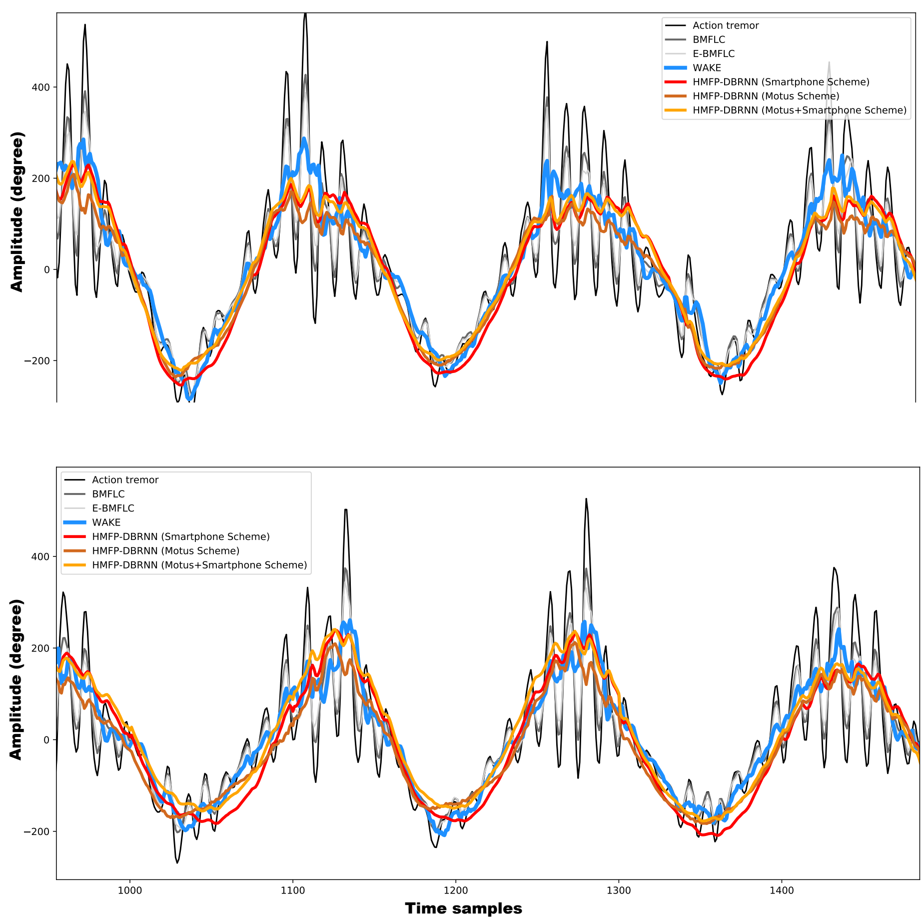


Figure 4.23: Performance of HMFP-DBRNN trained via three different schemes over the action tremor recordings.

Chapter 5

Data-driven Methods for Discrimination of Neurological Disorders

5.1 Introduction

In the previous chapters, the contributions of the thesis towards developing efficient data-driven models for BCI systems and ADs were presented. In particular, we introduced an enhanced classification scheme for multiclass problems, and an optimization framework to derive subject-specific spatio-spectral filters for EEG-based BCI systems. Thereafter, we introduced data-driven frameworks to process the PHT recordings from patients affected by neurological disorders to estimate and extract the voluntary component of motion from their tremorous hand movements. To further enhance the quality of assistive services delivered to patients, we believe that employing a disease-specific approach in tailoring the parameters of ADs for different neurological disorders would be a promising direction to investigate. Towards this goal, this chapter presents an innovative data-driven model, referred to as the NeurDNet, for accurate and efficient classification of PD and ET patients via hand motion recordings. The NeurDNet takes advantage of a 2-stage classification paradigm incorporating a DL core and a classical ML core, to accurately distinguish and classify the recordings of patients with PD and ET. In addition, the chapter introduces a novel objective function for a set of learning methods, referred to as metric learning, which are often employed in applications where limited number of examples from each class are available. Metric learning offers promising application in integrating ML techniques in biomedical problems, where due to the confidentiality and privacy constraints, large datasets of different diseases are not always accessible. Our proposed objective function, which we refer to as MPCL, offers a faster convergence rate and a more inclusive generalization over limited number of training examples from each class.

5.2 The NeurDNet Framework

NeurDNet is developed based on a unique, large, and inclusive dataset of hand kinematics, that is clinically collected in this study, which includes 250 tremor assessments of 81 patients. Each tremor assessment consists of recordings in 3 channels from 6 tasks in 3 trials, which together add up to 54 single-channel tremor recordings. The utilized dataset was collected at the London Movement Disorders Centre laboratory over a time span of 4 years. The comprehensive employed dataset of hand motion recordings has provided NeurDNet with the unique capability of perfectly magnifying and mastering the overlapping features of the two disorders (i.e., PD and ET), hence, decreasing the misdiagnosis error and maximizing the classification accuracy. The exceptionally large and inclusive dataset enables NeurDNet to reliably capture the underlying and overlapping features of the two diseases and provides an acceptable degree of generalization to the network.

- A novel data-driven architecture, i.e., NeurDNet is developed and trained over a large and comprehensive dataset of hand kinematics collected over a time span of 4 years and consisting of about 90 hours of recordings from 81 patients. This dataset has captured the acceleration of hand motion in PD and ET patients in 3-axes, while performing 7 different tasks in 3 trials, by mounting a triaxial accelerometer on the dorsum of their hand.

- The processing pipeline of NeurDNet is a sequential architecture of a CNN core and a classical ML core, which together form a two-stage classification paradigm for differential diagnosis. This novel architecture further boosts the reliability and accuracy of the system in diagnosing neurological disorders.

- To maximize the amount of extracted information from the dataset with the ultimate goal of maximizing the overall classification accuracy, in addition to the raw accelerometer signals, shortcut bits are also introduced to the deep neural architecture of NeurDNet to convey some information about the task associated with the tremor recording. This is critical, since different tasks would stimulate different characteristics of tremor in PD and ET patients. In other words, the label of each task performed by each patient is embedded as a hint vector in the final classification layer of the neural network to further boost the classification accuracy of NeurDNet in distinguishing the two diseases. As a result, patients should conduct a particular series of motion tasks (explained later) to activate different PHT patterns, which can be decoded into differential diagnosis using NeurDNet.

- Another major novelty of NeurDNet is employment of specialized and sophisticated methods in interpreting its decisions by explaining the clues in the input signals that lead to a particular class label. Such comprehensive analysis provides statistically significant and clinically viable knowledge for classification of PD and ET and relaxes the concerns on learning structural and unwanted biases in the input data that can lead to proper discrimination of the two diseases.

The above-mentioned contributions of NeurDNet collectively have resulted in the state-of-the-art mean classification accuracy of 95.55%. In the rest of this section, the data collection procedure for the employed dataset as well as the architecture of the NeurDNet framework and the rationales behind its design are discussed. Lastly, the evaluation metrics and the algorithmic workflow of NeurDNet are explained.

5.2.1 Data Preparation

To develop the NeurDNet framework, the dataset introduced in Section 4.3.1 is utilized. To prepare the data for this work, each tremor assessment is decomposed into 3 trials, 6 tasks, and 3 channels, contributing to preparation of a large collection of $13,500$ tremor signals. It should be highlighted that based on the validation results and the achieved classification accuracy in different scenarios, to develop and evaluate NeurDNet framework, the action tremor recordings associated with the "finger-to-nose" task are omitted from the dataset. As the action tremor recordings contain dynamic features from both the person's voluntary movement and the tremorous movements, we believe that the wide range of characteristics and dynamic properties of the voluntary component misleads NeurDNet in the classification tasks and degrades its accuracy. To develop the first-stage classifier and identify the hyper-parameters of neural network, the dataset is split based on $[75\% - 25\%]$ portions for training and testing, where the 5-fold cross-validation is performed using the samples in the training set. It is worth noting that the two sets are formed based on subjects and the recordings of one subject only contribute to one set, as an attempt to eliminate any direct or indirect leakage of information from the training set into the test set. Once the hyper-parameters of the first-stage classifier were determined, the second-stage classifier was added to the system and the whole pipeline was trained and evaluated for different training/test proportions. In other words, $[61,20]$, $[54,27]$, $[46,35]$, $[38,43]$, $[30,51]$, and $[22,59]$ number of patients are employed respectively to form [training, test] sets in 25%, 35%, 45%, 55%, 65%, and 75% cases. It is also worth mentioning that the whole process of fine-tuning the hyper-parameters of the first-stage classifier is based on the mean accuracy of the classification in the cross-validation process and the test set is only employed to perform the final evaluation of NeurDNet, as shown in the results reported in Tables 5.1 and 5.2.

Prior to utilizing the recordings for training and evaluation stages, the entire tremor signals were downsampled to 100 Hz to minimize the computational burden on the system as well as the complexity of the network. It should be noted that as the informative spectral region in the tremor signals spans the range up to 20 Hz, and according to the Nyquist theorem that sampling a signal with at least twice the rate of its maximum informative frequency is enough to fully reconstruct it, we believe that 100 Hz is low enough to avoid excessive computational costs on the system and high enough not to distort the spectral contents of interest in the signal. Afterwards, the mean of each signal is subtracted from itself to eliminate the effect of calibration and the bias associated with the posture of each task. Finally, the spectrogram of each tremor signal, which is of 20 seconds length, is calculated according to the Welch method, by moving a Hamming window of length 100 points, the overlap size of 90 points, and the FFT resolution of 256 points. Each downsampled tremor assessment is then represented by a $2-$dimensional matrix of size $[129 \times 191]$. As shown in Figure 5.1, the obtained spectrograms of the tremor signals are then fed to NeurDNet to be processed by the convolutional layers of the first-stage classifier.

5.2.2 Architecture of NeurDNet

To diagnose each patient, NeurDNet takes advantage of a two-stage classification paradigm, which is designed to collectively employ the information stored in the time-series recordings of each patient, as well as their behavioral patterns in different tasks. Each tremor assessment consists of recordings in 3 channels from 6 tasks in 3 trials, which together add up to 54 single-channel tremor recordings. The first-stage classifier is designed to vote for each of the single-channel recordings, whether they are PD or ET. When 54 votes for a tremor assessment are collected, the class labels or probabilities associated with each class are fed to the second-stage classifier. We believe that the two stage classification paradigm enables us to extract the underlying and discriminating patterns of tremor signals as well as the discriminating behavioral patterns of patients in case of performing different tasks.

The first-stage classifier takes advantage of convolutional neural architectures to process the spectrogram representations of the single-channel recordings. As shown in Figure 5.2(a), 2 convolutional layers followed by 3 dense layers build up the first-stage classifier. The details of the convolutional layers are given in the figure. The first dense layer employs ReLu activation functions and the second one employs Leaky-ReLu with the parameter of 0.1 as its activation function. A crucially important and novel characteristic of the designed first-stage classifier is employing shortcut bits for the second dense layer to introduce the origin of the input signal to the network. In other words, along with the spectrogram of a tremor signal, a binary vector of 6 bits is directly concatenated with the output of the first dense layer to form the input to the second dense layer. This vector encodes each clinical task with a binary vector and provides the network with extra information to conclude the label of a tremor signal. To train the network, the mean softmax cross entropy between the output of network and the true labels is minimized by employing Adam Optimizer with the learning rate of 0.0001. Performance monitoring over the validation set revealed that 44 epochs of training reach an optimal point in the learning curve and thus, the training process is stopped after 44 epochs. The maximum accuracy achieved only on the first-stage classifier is 75.55% over the validation set. It is worth noting that it is good practice to evaluate the framework only when the development phase is finalized and the whole processing framework (NeurDNet) is ready to be assessed on the test set. As such, for the first-stage classifier there is no choice other than reporting its performance over the validation set.

The second-stage classifier, on the other hand, is developed based on classical classification techniques and the maximum accuracy of 95.55% is achieved when Quadratic Discriminant Analysis (QDA) technique is applied on the outputs of the first-stage classifier. As shown in Figure 5.2(b), the votes of the first-stage classifier for one tremor assessment (54 votes for each tremor assessment) are collected in terms of probabilities for each class and a feature vector of length 54 is formed to train/evaluate the QDA classifier. To classify an unlabeled tremor assessment, the 54 features associated with it are derived to form the feature vector $\boldsymbol{f}$. The classification is based on the prior probability of the classes given the feature vector, i.e., $p(y = class|\boldsymbol{f})$, as such, according to the Bayes' theorem, the posterior probability of $p(\boldsymbol{f}|y)$ needs to be calculated. In QDA classifier, the posterior probability is modeled as a multivariate Gaussian distribution, and thus, a likelihood ratio for the two

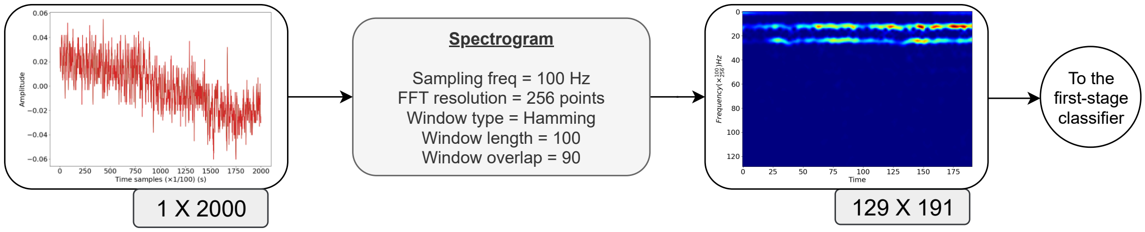


Figure 5.1: The preprocessing step to convert time-series tremor recordings into 2D spectrotemporal representations of the signals to be processed with the first-stage classifier of NeurDNet.

classes given the feature vector and the information from training samples is calculated as

$$\text{Likelihood ratio} = \frac{\sqrt{2\pi\boldsymbol{\Sigma}_{PD}}^{-1}\, exp\Big(-0.5(\boldsymbol{f}-\boldsymbol{\mu}_{PD})^T\boldsymbol{\Sigma}_{PD}^{-1}(\boldsymbol{f}-\boldsymbol{\mu}_{PD})\Big)}{\sqrt{2\pi\boldsymbol{\Sigma}_{ET}}^{-1}\, exp\Big(-0.5(\boldsymbol{f}-\boldsymbol{\mu}_{ET})^T\boldsymbol{\Sigma}_{ET}^{-1}(\boldsymbol{f}-\boldsymbol{\mu}_{ET})\Big)}, \tag{5.1}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ respectively represent the mean and covariance matrix of features for the PD and ET classes.

5.2.3 Hyper-parameter Optimization of NeurDNet

In the validation process, all of the parameters and hyper-parameters of NeurDNet are fine-tuned to maximize the classification accuracy. To fine-tune the hyper-parameters of the first stage classifier, which is a CNN-based deep neural model, the hand motion dataset is strictly split into 2 sets, 75% for training and 25% for testing. To avoid the leakage of information from the training set to the test set, the formation of datasets is based on tremor assessments from patients and the recordings of each patient are only participated in one set. This strategy is used to impose harsh evaluation conditions on NeurDNet to better investigate its capability in extracting the generic underlying patterns of each disease from the hand motion recordings. To identify the optimum hyper-parameters of the network and validate its performance over different hyper-parameters, a $5-$fold cross-validation procedure is employed over the samples in the training set. In fact, each round of training is performed over $4/5$ of the training set and the rest of the samples are utilized for validation and this process is repeated for 5 times with completely exclusive validation samples. Finally, the mean performance over the 5 runs is reported as the accuracy of network for the selected hyper-parameters. In addition, cross-validation enables us to decide if the model is overfitted to training samples or not and investigate if the network generalizes well over the wide and overlapping range of hand motion characteristics for the two diseases. The hyper-parameters of the network are altered to achieve the maximum validation performance over the range of searched hyper-parameters. To summarize, 75% of data ($10,125$ samples) is used for training, 25% ($3,375$ samples) is reserved for evaluation. The best classification accuracy of the first-stage classifier over the validation data is 75.55%. It should be noted that each tremor assessment consists of 54 tremor signals (6tasks $\times$ 3trials $\times$ 3channels) and the above-mentioned accuracy is achieved for classification of each tremor signal, therefore, the achieved performance does not reflect the accuracy of the NeurDNet on classifying the "patients" or "tremor assessments" into PD or ET.

Upon fine-tuning the best hyper-parameters for the first-stage classifier through a rigorous grid-search procedure, a similar strategy is followed to identify the best hyper-parameters for the second-stage classifier. To develop the second-stage classifier, the first-stage is kept intact based on the best derived hyper-parameters and the second-stage classifier is updated to achieve the best performance, i.e., highest classification accuracy. The second-stage classifier can be characterized by two main hyper-parameters, i.e., the type of input features and the classification methodology. As the second-stage classifier is fed with the output of the first-stage block, the output of the first-stage classifier could be obtained either in binary format (class labels) or numeric format (probability associated with each class). The classification scheme of the second classifier is another hyper-parameter that its effect is investigated on the overall performance of NeurDNet. To this aim, a set of classifiers with different settings are employed to be coupled with the first-stage classifier. The evaluated paradigms include RF, SVM, NB, LR, AB, LDA, QDA, DT, and MLP. Please note that the parameters defined in the parentheses of the first column of Table 5.1, indicate the option in which the classification algorithm is employed; "*Entropy*" and "*Gini*" define the clustering criteria for RF or DT, "*Radial Basis Function (RBF)*" and "*Linear*" define the type of kernel used by SVM, "*Singular Value Decomposition (SVD)*" and "*Least Squares Error (LSQR)*" indicate the eigenvalue solver for LDA, and "*MLP(N)*" defines a one layer neural network with *N* nodes.

Finally, to check the sensitivity of NeurDNet to the amount of available training data and its capability to infer the underlying characteristics of the two diseases from the recordings, the performance of network is trained and evaluated across different choices for test set population, which are 25%, 35%, 45%, 55%, 65%, and 75% of the whole dataset. Please note that the aforementioned ratio indicates the portion of dataset to form the test set. In addition, it should be noted that to decrease the effect of randomness in selecting the train/test subjects, each evaluation is performed for 30 times and the mean accuracy of this comprehensive performance evaluation is reported in Table 5.1. As it is observed, the maximum classification accuracy is obtained when QDA classifier is coupled with the first-stage classifier, 75% of dataset is employed for training purposes, and probabilistic feature vectors are fed to the second-stage classifier. The second best accuracy is also obtained in similar settings, except for the case that 65% of dataset is employed for training. It is worth mentioning that the accuracy of the second-stage classifier is actually the accuracy of NeurDNet in classifying the two diseases and is obtained by processing the whole tremor assessment of a subject, i.e., 54 tremor recordings from 6 tasks, in 3 trials, and in 3 channels.

5.3 Simulation Results

In this section, the NeurDNet framework is evaluated based on several different test paradigms and the results are presented. As thoroughly discussed in the Methods section, the best classification accuracy of NeurDNet is achieved when a CNN architecture, as shown in Figure 5.2(a), is used as the first-stage classifier and the outputs of the CNN model for each

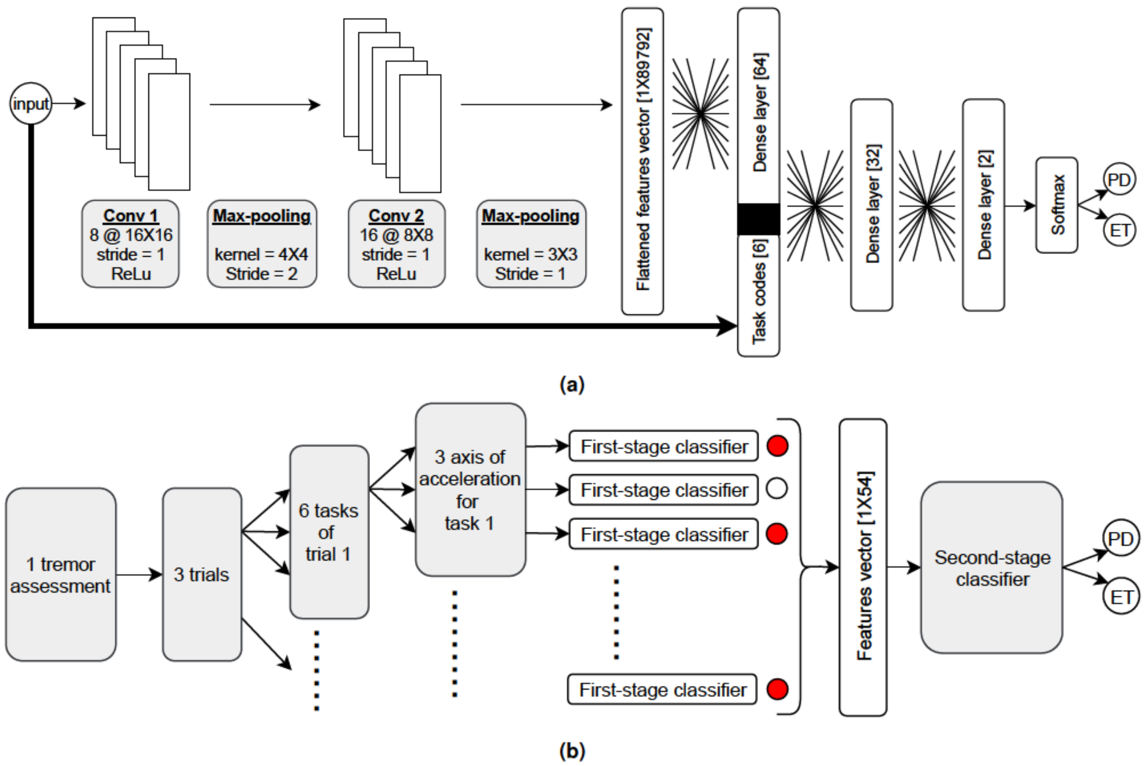


Figure 5.2: The overall processing framework of NeurDNet to perform differential diagnosis between PD and ET patients. **(a)** This part depicts the processing pipeline for the first-stage classifier, which is based on Convolutional Neural Networks. In this stage, a preliminary decision (PD or ET) is made on a single signal of tremor assessment, which is previously passed through the preprocessing block. This signal could be the acceleration of hand motion in any axis, from any task of any trial. **(b)** This figure shows the second stage of the classification process for each tremor assessment. In fact, each tremor assessment contains 54 tremor signals, where all of them are passed through the first-stage classifier. Then, the decision on each signal is aggregated in a vector of length 54 which forms the feature vector for the second-stage classifier.

tremor assessment are fed to a QDA model as the second-stage classifier, as shown in Figure 5.2(b). Each tremor assessment constitutes of 54 single-channel tremor signals and the role of the first stage classifier is to classify each of these signals into PD or ET. Then, the collection of 54 predictions is fed to the QDA classifier as a feature vector, and the final vote for each tremor assessment is obtained by the second-stage classifier. It should be noted that the best classification accuracy, which according to Table 5.1 is 95.55%, is achieved when the training/test ratio of 3 : 1 (75% of data is reserved for training) is followed.

To further investigate the performance of NeurDNet, the confusion matrix and the Receiver Operating Characteristics (ROC) curve for the winning frameworks of NeurDNet is obtained. ROC curve helps us understand the diagnostic capability of a binary classifier by measuring the sensitivity and specificity of classification for different thresholds of distinguishing the two diseases. To define the meaning of sensitivity and specificity in this context, first, the terms of “positive” and “negative” diagnosis need to be defined. Basically, the term “negative” stands for healthy diagnosis of an individual and the term “positive”

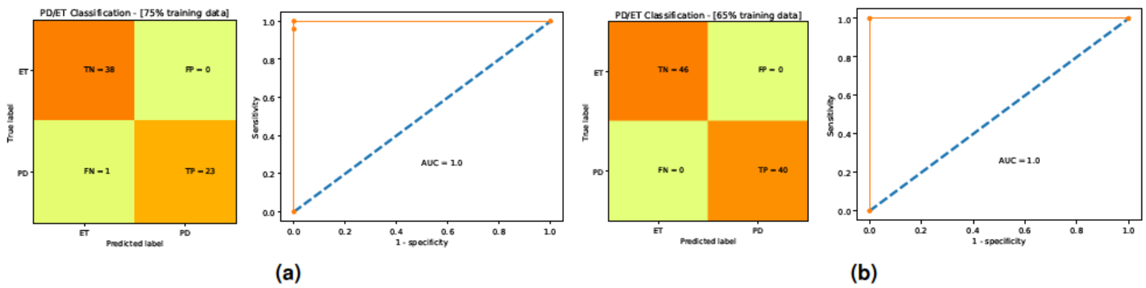


Figure 5.3: Confusion matrix and ROC diagrams associated with the 2 winning frameworks for PD/ET classification. Please note that AUC stands for Area Under Curve. Two winning paradigms of NeurDNet are when QDA classifiers is coupled with the first-stage classifier and **(a)** 75% and **(b)** 65% of the dataset is used for training process, respectively.

stands for the opposite. However, as in this work we are not dealing with a healthy/patient problem and our goal is to distinguish between the two diseases, we redefine the terms "positive" and "negative" as being diagnosed as PD and ET, respectively. Thus, the sensitivity (specificity) of NeurDNet is the ratio of the correct PD (ET) classifications over the total number of PD (ET) cases. ROC curve illustrates *sensitivity* against $(1 - specificity)$ and helps physicians to choose a proper threshold to attain a certain degree of sensitivity or specificity. In addition to determining the classification threshold, another important classification measure that is derived based on the ROC curve is the "Area Under the Curve (AUC)" criteria. AUC indicates how well a classifier distinguishes two classes and its value in the range between 0.5 to 1 reflects the performance of the classifier from "no discrimination capacity" to "perfect discrimination capacity", respectively. To obtain the confusion matrix and ROC curves for NeurDNet, the two most accurate classification paradigms in Table 5.1 are selected and the results are shown in Figures 5.3(a) and 5.3(b). It is worth mentioning that to generate the plots in Figure 5.3, the output of a complete classification pipeline with fixed training and testing set needs to be analyzed, however, the reported values in Table 5.1 are obtained by averaging over 30 trials, thus the mean value is not necessarily associated with any of the 30 random runs. It is worth mentioning that to generate the plots in Figures 5.3(a) and 5.3(b), the training set that leads to maximum classification accuracy among the 30 random formations of the train and test sets is utilized.

5.3.1 Explainability of NeurDNet

Generally speaking, the capability to identify and explain the internal process that leads to a certain outcome is referred to as the *explainability* of machine learning models (XAI), which plays an important role in approving the applicability of model and reliability of its results. When it comes to employing deep neural networks in biomedical domain, due to the sensitivity of application and the risk of fatal errors, the explainability of the model becomes of much greater importance. In this section, the explainability of NeurDNet s investigated by extracting clues in the tremor signals that are important and noticeable in concluding the label of an unseen tremor assessment. In other words, the regions in the spectra-temporal representation of tremor recordings are discovered, such that motivate

Classifier	**Binary Features**						**Probabilistic Features**					
	25%	**35%**	**45%**	**55%**	**65%**	**75%**	**25%**	**35%**	**45%**	**55%**	**65%**	**75%**
RF (entropy)	85.69	84.24	82.91	81.94	82.43	78.68	86.18	85.43	83.79	82.66	82.20	78.21
RF (gini)	85.43	84.59	83.43	82.35	81.97	78.28	86.49	84.81	84.27	82.63	82.57	78.29
SVM (rbf)	85.68	84.65	84.24	82.19	83.10	79.46	86.33	85.83	85.38	82.09	82.68	79.01
SVM (linear)	84.26	82.69	82.08	81.34	80.78	78.02	85.83	84.77	83.60	82.36	82.02	78.57
NB	83.70	83.55	80.23	81.44	81.67	77.31	85.98	86.42	84.94	83.94	84.15	81.48
LR	85.76	84.41	84.09	83.10	82.83	79.49	**87.29**	86.10	85.28	83.65	83.38	79.74
AdaBoost	83.97	81.61	80.99	79.95	79.30	75.80	85.03	82.97	81.53	80.01	78.12	73.32
LDA (svd)	79.54	76.25	75.83	73.79	66.21	67.44	77.81	76.41	76.56	72.31	65.12	63.62
LDA (lsqr)	79.54	76.25	75.80	73.77	63.40	49.57	77.81	76.41	76.56	72.31	65.12	49.50
QDA	81.85	83.18	78.69	72.08	63.26	58.62	**95.55**	**93.89**	81.73	73.48	56.29	53.13
DT (entropy)	81.21	78.45	77.66	77.63	76.02	74.75	80.40	79.01	77.11	77.57	75.06	71.73
DT (gini)	80.45	80.16	78.51	77.25	77.32	75.25	77.99	78.29	76.89	76.35	74.29	71.84
MLP (10)	85.01	82.40	82.05	81.25	79.79	77.53	84.33	83.03	81.64	80.25	80.04	77.04
MLP (30)	84.64	82.84	82.02	80.85	79.63	77.49	84.53	82.80	81.79	80.50	80.33	77.45

Table 5.1: Classification accuracy of NeurDNet in the two cases of employing binary and probabilistic features. The classification accuracy is measured across different choices of the second-stage classifier, including Random Forests (RF), Support Vector Machines (SVM), Naive Bayes Classifier(NB), Logistic Regression (LR), AdaBoost Classifier (AB), Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Decision Trees (DT), and Multi Layer Perceptron (MLP).

the network to select one class over the other. To this aim, the "Gradient-weighted Class Activation Mapping (Grad-CAM) [250]" methodology is employed to discover the parts of the input spectrogram to the CNN, which contribute to assignment of a label to the input. To obtain the Grad-CAM representations of NeurDNet, as shown in Figure 5.4, all the output nodes of the first-stage classifier are set equal to zero except the one that corresponds to the correct label. Then, the gradients of this output are backpropagated to the network and a heatmap mask for the input signal is obtained. The mask assigns a weight to each pixel of the input signal to determine its importance in concluding the final label. To implement this process, the "keras-visualizations" library [251] in Python coding language is employed.

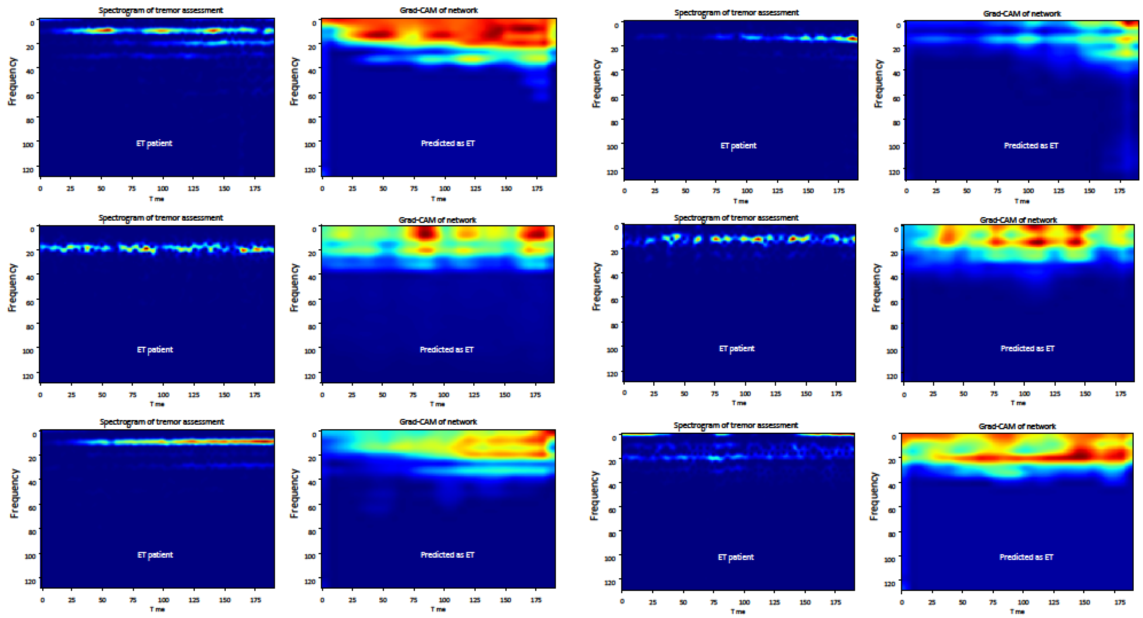
Although the Grad-CAM plots in Figure 5.4 clearly represent the important regions for NeurDNet to distinguish PD from ET and provide some interpretable insights on the valuable spectral contents for each of the two diseases, we need to statistically validate our observations over the whole dataset and investigate if the observed differences are meaningfully valid for all of the samples in the dataset. To this aim, the heatmaps for the two classes are statistically and in a pixel-wise fashion compared to check if any pixel takes significantly different values for the two classes. In this regard, first, the Grad-CAM representation of NeurDNet for all of the PD and ET tremor assessments that are correctly classified is derived. Due to the large population of instances for each group ($3,788$ and $2,473$ for ET and PD, respectively), the "$z-test$" needs to be employed to check if the mean of the Grad-CAM analysis for each pixel and of the two groups is significantly different. As the $z-test$ procedure is based on normal distribution of data, first, the D'Agostino and Pearson's test [249] was performed to verify the normality of data. Once this condition was relaxed, which was expected due to the large number of instances and insights from the central limit theorem, the element-wise $z-score$ across the whole dataset and between the two groups was derived as

$$z = \frac{\bar{g}_{ET} - \bar{g}_{PD}}{\sqrt{\frac{\sigma_{ET}^2}{n_{ET}} + \frac{\sigma_{PD}^2}{n_{PD}}}}, \tag{5.2}$$

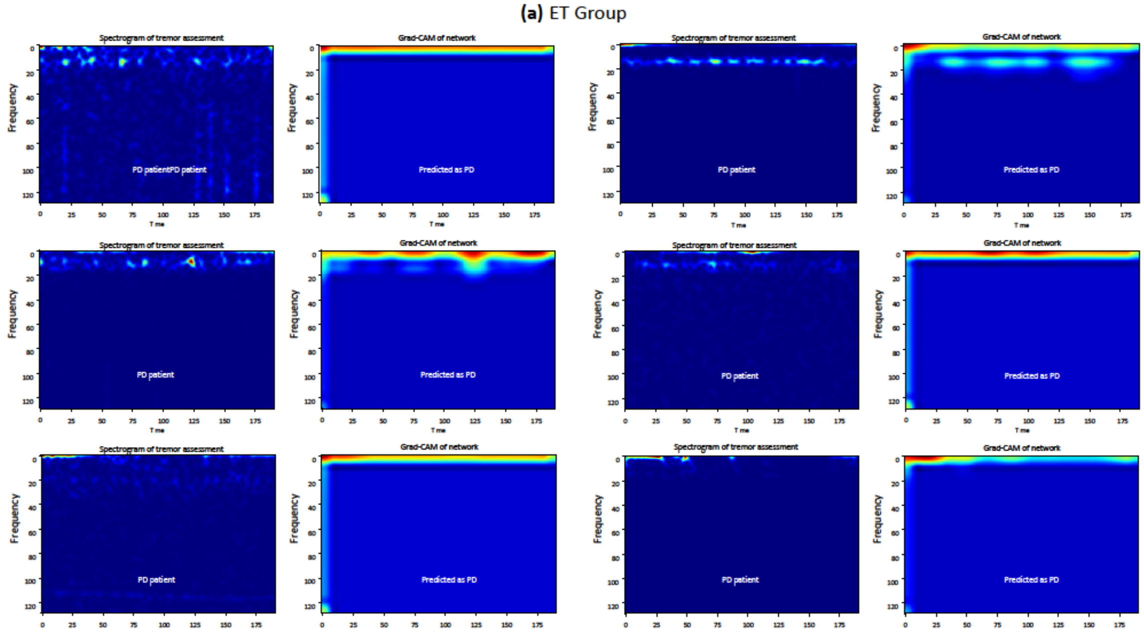
where $\bar{g}_{ET}, \bar{g}_{PD}, \sigma_{ET}^2, \sigma_{PD}^2, n_{ET}$, and n_{PD} are the mean value of Grad-CAM pixel for ET group, mean value of Grad-CAM pixel for PD group, variance of the pixel across ET group, variance of the pixel across PD group, population of ET samples, and population of PD samples, respectively. Please note that $z-score$ is calculated for each pixel across the two groups and the $p-value$ is respectively obtained for each pixel. According to the formulation of $z-score$ in Eq. (5.2), positive $z-score$ corresponds to higher attention of NeurDNet to ET features and the opposite stands for the PD group. Thus, to derive the masks associated with each group, the area under a standard normal distribution, $auc(z_0)$ was derived as in

$$auc(z_0) = \int_{-\inf}^{z_0} p_Z(z)dz, \tag{5.3}$$

where $Z \sim \mathcal{N}(0,1)$ or in other words $p_Z(z) = \frac{1}{2\pi}\exp(-z^2/2)$. For the ET group, the mask is of the same dimension as the input spectrogram and is a zero matrix, except for the pixels that $auc(z) > 0.99$. Conversely, for the PD group, the mask is obtained by selecting the pixels for which the $auc(z) < 0.01$. In fact, this process is equal to selecting the pixels where the Grad-CAM analysis of NeurDNet shows significantly different means for the two groups by setting $\alpha = 0.02$ ($p - value < 0.02$). Afterwards, the masks are applied on the mean Grad-CAM representation of NeurDNet for PD and ET groups to reveal the important temporal and spectral regions for classification of each group. The results of this analysis are shown in Figure 5.5.



(a) ET Group



(b) PD Group

Figure 5.4: Analysis of explainability for NeurDNet. It should be highlighted to convert the values *y-axis* scale to frequency in Hz, the values need to be multiplied by 100/256.

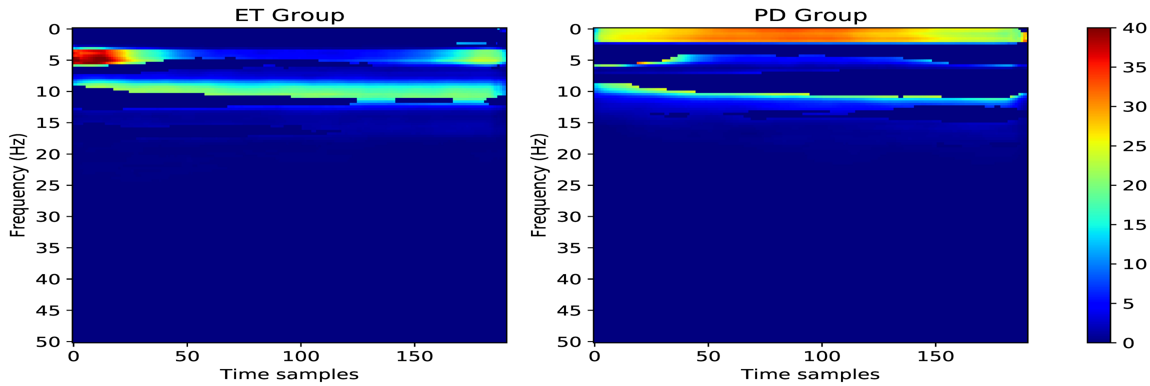


Figure 5.5: Results of the statistical test over the Grad-CAM analysis of NeurDNet for the two diseases. The intensity of different parts in the spectrogram determines the importance of the region for NeurDNet to conclude the class of the tremor assessment.

5.3.2 Analysis of the Dominant Features

Similar to the previous section where the learned features of the first-stage classifier through the Grad-CAM analysis were investigated, in this section, we identify the importance of task-specific features to classify the tremor assessments in the second-stage classifier. The results presented in this subsection are obtained by analyzing the winning architecture of NeurDNet, which is trained over 75% of the dataset and employs probabilistic features with QDA classifier. To identify the role of each feature in forming the final decision of NeurDNet for an input signal, a sequential and iterative feature selection approach, referred to as the wrapper method, is employed. In this technique, the classification accuracy for different subsets of features is calculated and the subset with the highest classification accuracy contains the most influential features. In addition, the wrapper method does not utilize similarity or scoring criteria to compare the features with labels; instead, the dominant features are selected based on their effect in the final classification accuracy. In this work, to discover the efficacy of each feature, the best feature through the discussed sequential process is selected, then it is removed from the pool of features, and then again the best feature in the pool is selected. This process continues until all of the features are drawn from the pool and all of the features are sorted based on their role in forming the final decision of the classifier. The results of this process are shown in Figure 5.6.

5.3.3 Accuracy of the First-visit Diagnosis

Another characteristic of NeurDNet, which is crucially important from a clinical point of view, is the accuracy of NeurDNet in diagnosing patients in their first visit to the clinic. As previously discussed, a number of tremor assessments in the employed dataset are collected from patients in their second visit to the clinic, in 6 weeks after their first visit. Until now, the reported performances of NeurDNet are based on collectively processing the tremor assessments from the first and the second visits, which might be biased due to presence of any identifiable or unidentifiable role playing factor between the two visits. In other words, some factors like the familiarity of patients with the tasks, and the effect of any potentially

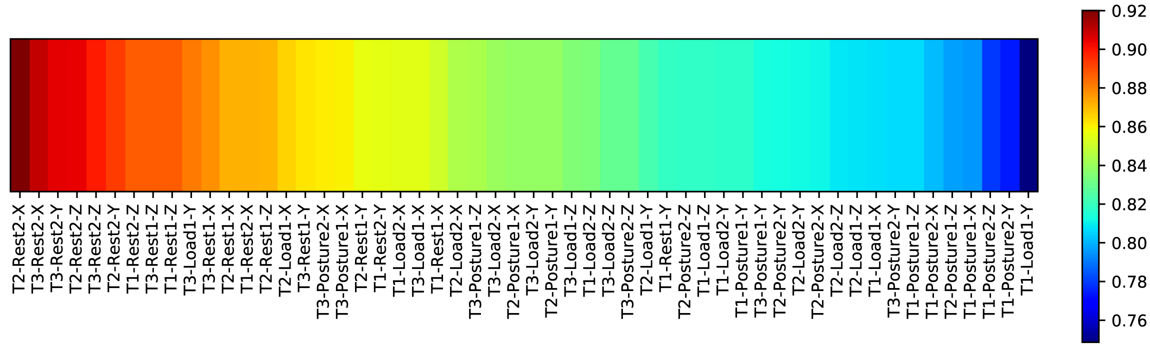


Figure 5.6: Results of sequential feature selection for the features that are fed to the second-stage classifier. Please note that these results are obtained through a 5−fold cross-validation process, when 75% of dataset is used for training. It should be highlighted that in this analysis, the probabilistic features due to their superior performance over binary features are employed, and the label of each feature is formed as [TrialNumber-TaskName-RecordingChannel].

received medication within the 6 weeks period can change the distribution of input data between the two visits, and may leave a positive or negative impact on the performance of NeurDNet. From a clinical standpoint, it is imperative to conclude a correct diagnosis in the first visit to minimize the side-effects associated with misdiagnosis. The results of this analysis are provided in Table 5.2. As it is understood from the results, the maximum classification accuracy of 93.05% is achieved when QDA classifier is coupled with the first-stage classifier and 25% of dataset is reserved for evaluation. It is worth reiterating that the formation of the training and evaluation sets is based on the subjects and all of the tremor assessments from one subject contribute only to one set, even if the patient has revisited the clinic in 6 weeks. Another point with regards to Table 5.2 is that the reported classification accuracies are obtained via a Monte Carlo simulation technique, i.e., averaging the classification accuracy of NeurDNet over 30 random formations of the training/evaluation sets.

5.3.4 Discussion

The results presented in Table 5.1 clearly suggest that the maximum classification accuracy is achieved when QDA classifier with probabilistic features are employed and the whole system is trained over 75% of the dataset. In addition, it is worth highlighting that the consistency of results for different training/test ratios is also an important measure for robustness of a framework and reveals the capability of the NeurDNet framework in generalizing over the underlying patterns of the studied phenomenon. Based on this argument, we can also nominate the Naive Bayes classifier as a successful classification method to be coupled with the first-stage classifier. The consistency of results for the NB classifier with probabilistic features across different training/test ratios, even for the minimum value of 25% for training (75% for evaluation), reveals the superior capability of this classifier in grasping the overall distribution of features for the two PD and ET classes. The observed behaviour of the NB classifier in this work is also consistent with its renowned capability in extracting strong classification rules based on minimum amount of training data.

Classifier	Binary Features						Probabilistic Features					
	25%	**35%**	**45%**	**55%**	**65%**	**75%**	**25%**	**35%**	**45%**	**55%**	**65%**	**75%**
RF (entropy)	87.31	85.30	83.66	81.90	81.43	79.60	86.78	86.13	84.78	82.36	81.53	81.05
RF (gini)	87.59	85.80	83.50	82.03	80.96	79.77	86.66	85.63	84.83	82.23	81.38	80.83
SVM (rbf)	87.05	85.89	84.51	82.07	81.45	78.66	**88.26**	86.50	86.13	82.81	82.22	79.85
SVM (linear)	85.85	82.56	82.47	81.15	79.81	77.90	86.82	84.86	83.83	82.39	81.34	80.14
NB	84.99	83.93	79.95	81.65	77.07	75.61	87.60	86.44	85.11	84.54	82.57	81.09
LR	87.43	85.26	84.05	81.88	80.92	78.78	88.08	86.62	86.30	83.52	82.74	80.94
AdaBoost	86.26	82.53	82.70	80.21	77.69	76.46	85.79	83.59	82.32	80.78	78.63	75.06
LDA (svd)	81.13	78.10	76.10	70.13	67.04	67.82	79.12	77.02	76.49	71.58	66.14	62.99
LDA (lsqr)	81.13	78.10	76.04	70.13	62.49	49.41	79.12	77.02	76.49	71.58	66.14	51.06
QDA	79.18	80.77	77.65	70.20	62.15	60.04	**93.05**	**89.66**	77.59	71.63	59.92	54.01
DT (entropy)	80.85	79.04	78.25	76.60	76.42	74.96	79.76	79.14	78.44	77.39	75.61	73.82
DT (gini)	81.78	80.40	78.63	76.86	75.54	73.97	80.35	77.90	78.09	77.47	76.28	74.15
MLP (10)	85.41	83.33	82.54	78.90	78.09	77.60	83.31	81.84	81.50	79.72	78.15	77.83
MLP (30)	85.74	82.76	81.48	79.00	78.42	77.23	83.80	82.24	81.87	79.70	78.22	77.55

Table 5.2: Classification accuracy of NeurDNet when only the first-visit tremor assessments are included in the test set. The classification accuracy is measured across different choices of second-stage classifier, including Random Forests (RF), Support Vector Machines (SVM), Naive Bayes Classifier(NB), Logistic Regression (LR), AdaBoost Classifier (AB), Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Decision Trees (DT), and Multi Layer Perceptron (MLP).

The plots in Figure 5.5 clearly represent the informative regions in the spectrotemporal plots of tremor signals, which are insightful for the diagnosis of patients. As it is observed, PD is mainly characterized by occurrence of low frequency vibrations on the hand motion, whereas ET is mainly characterized by high spectral activity in the hand motion signals. The highlighted regions for each disease are also compatible with their physiological characteristic, where ET is known to occur in a wider spectral range than PD. It should be noted that the highlighted regions in the mean spectrotemporal maps of ET and PD populations in Figure 5.5 do not imply that the spectral contents of each disease are only stored in those areas. On the contrary, the highlighted regions identify statistically significant regions in the spectrotemporal map of signals, which provide informative and strong clues for the network and potentially for the physicians to discriminate the two diseases.

The plot in Figure 5.6 determines the efficacy of each tremor assessment task in providing useful information for differentiating of PD from ET, and reveals that the features obtained from the "Rest1" and "Rest2" tasks convey exceptionally valuable information to the classifier to discriminate between the two diseases. In fact, the plot shows the achieved classification accuracy when only one of the features is utilized to form the classifier. Since there is a pool of 54 features for each tremor assessment, investigating the efficacy of all possible subsets of features on the classification accuracy of NeurDNet would have imposed high computational burdens on the development phase (assessment of 2^{54} cases) and thus, the effect of features are examined based on a naive assumption where features are considered independently. As stated previously, a crucially important feature for a clinically applicable and reliable diagnostic procedure is its performance in correctly diagnosing the patients in their first visit to the clinic, as the side-effects associated with misdiagnosis could be minimized and also the patients have a higher chance receiving medications and treatments in the earlier stages of the diseases. The presented results in Table 5.2 confirm that the NeurDNet framework achieves a high classification accuracy over the tremor assessments recorded in the first visit of patients to the clinic.

Here, to better highlight different aspects of the NeurDNet in terms novel features, comparisons with prior works, and its advantages and disadvantages, the following itemized discussion is provided.

- ***Novel Features of the NeurDNet Framework:***
  - NeurDNet produces a novel and accurate machine intelligence pipeline designed based on a particular collection of hand tremor to conduct a differential diagnosis between PD and ET.
  - Using the state-of-the-art Grad-CAM analysis, this paper, for the first time, highlights segments of the spectrotemporal behavior of hand tremors which has the most discriminative power for conducting PD versus ET diagnosis using advanced, recurrent neural network approaches. This makes the proposed information processing pipeline an explainable model (under XAI), which is the new generation of machine intelligence, tangential to the conventional blackbox implementation, which does not provide any insight on the decisions made and was susceptible to biases in the datasets. This major novelty of the proposed

NeurDNet is highlighted as the comprehensive study and analysis over the explainability of network and the corresponding statistical analysis conducted in the paper over the clues in the input signals leading to certain labels (PD or ET). The results of this analysis not only provide clinically viable information on the clues to discriminate PD from ET, but also relax the concerns on learning the structural and unwanted biases in the input data that can take part in discriminating the two diseases.

  - NeurDNet introduces a sequential processing pipeline based on a CNN core and a QDA classifier, which offers a multi-stage classification paradigm for differentiation of PD from ET. This unique architecture enhances the reliability of the system in determining if the unseen patients are PD or ET by analyzing the dynamics of hand in a hierarchical format.
  - The processing framework of NeurDNet is intelligently designed to maximize the amount of information exploited from the dataset by not only processing the signals representing the dynamics of hand motion but also incorporating the task labels to further assist the framework in interpreting the signals. This novelty in the architecture of the neural network catalyzes the classification accuracy of NeurDNet.
  - The NeurDNet framework is developed over a substantially large dataset of hand dynamics containing 87.5 hours of PHT recordings from 81 PD and ET patients. In this unique dataset, the dynamics of hand motion are examined in 7 different scenarios, which further increase the amount of information obtained from limb movement in patients with PD or ET.

- ***Comparison with Prior Works:*** By revisiting the list of recent research works with potentially relevant objectives as NeurDNet in Table 1, it is readily understood that NeurDNet outperforms the state-of-the-art accuracy in discriminating PD from ET. It also offers a novel machine intelligence pipeline which can be interpreted from the clinical point of view. Considering the two predecessors of NeurDNet with the highest classification accuracies (before the invention of NeurDNET in this paper), i.e., References [167] and [141], it is understood that NeurDNet not only excels the classification accuracy of the research that is based on accelerometer data but also outperforms that is based on Electromyogram (EMG) signals recorded from a tremorous hand (which was supposed to have richer neurophysiological content in the signal). To be more specific, here we provide an itemized comparison with two research publications, leading the state-of-the-art classification accuracy for discriminating PD from ET.

  - The work in reference [141] has collected accelerometer data and offers a classification accuracy of 92%. This work uses a tremor stability index as the feature for classification of PD from ET, which is derived by performing spectral analysis over a signal of length 100 seconds. Through their experiments, a certain threshold value for the stability index is determined for classification. On the contrary, the classification strategy of the NeurDNet assigns a probability to the final label of an unseen patient by analyzing the acceleration of hand motion in different axes, different tasks, and different trials, which offers a higher classification accuracy and a much more robust diagnostic framework. In fact, the proposed strategy enhances the reliability of the system in diagnosing patients and grants it a great degree of generalization over the characteristics of hand tremor. Besides the fact that our proposed NeurDNet framework achieves a higher classification accuracy, we believe that our analysis over 87.5 hours of tremor recordings (compared to 2.527 hours in reference [141]) achieves a better generalization over the wide and overlapping range of features in hand tremor among PD and ET patients, and provides a more robust classification paradigm.

  – Another counterpart to the NeurDNet that achieved the state-of-the-art classification accuracy of 94% [167] employs a combination of EMG recordings from tremorous hand and a set of physiological features collected from 54 patients and introduces a classification framework based on decision trees. Comparing the devices, it can be mentioned that EMG studies are typically more complex, requires more rigorous calibration, and is more expensive, all of which would affect the scalability of the machine intelligence in clinics, especially for those who are not sophisticated. Our proposed NeurDNet framework not only has recruited more number of patients for its analysis, which leads to better generalization over the inter and intra-personal variance of features in the hand tremor but also only employs an accelerometer to collect the required signals, which is a more cost-effective, more scalable, and portable solution, and requires a very each calibration process when compared with EMG studies.

- ***Advantages and Disadvantages of NeurDNet:*** Besides the fact that our proposed NeurDNet framework achieves a higher classification accuracy, we believe that our analysis over 87.5 hours of tremor recordings (compared to 2.527 hours in reference [141]) achieves a better generalization over the wide and overlapping range of features in hand tremor among PD and ET patients, and provides a more robust classification paradigm. Also, our employed dataset examines the dynamics of hand in 6 different static positions, which further reveals the behavioral patterns of the hand tremor and minimizes the risk of overfitting in the framework. In addition, the NeurDNet is grounded on analyzing the accelerometer signals, representing the dynamics of hand motion in different axes, which compared to a considerable number of research works focused on differentiating PD from ET by means of EMG signals, offers a more cost-effective, accessible, and portable solution. It is worth highlighting that although the proposed NeurDNet requires a larger data collection from each patient, which might be tedious or boring for some patients, given the importance of correct diagnosis and the consequences associated with misdiagnosis of patients, we believe that NeurDNet is a more robust and reliable classification paradigm for the PD vs. ET problem. Above all, the NeurDNet, for the first time in this domain, presents a unique and comprehensive study of the explainability of the classification model, which is supported by a thorough statistical analysis of the results. This important feature, not only provides viable and statistically significant information for clinicians to discriminate PD from ET but also relaxes the concerns on the curse of overfitting to biases in the analyzed signals.

5.4 The MPCL Objective Function

In previous sections, the contributions of the thesis in developing data-driven computational models for various applications including BCI systems and ADs were covered. Although promising results and performances were achieved, a major bottleneck in fully leveraging the massive learning capacity of DL methods in medical domains is the data-hungry nature of such techniques and the unavailability of large-scale medical datasets. This section elaborates on another contribution of the thesis focused on employing a specific category of DL methods, referred to as Deep Metric Learning, for the problems with a limited number of samples per class. To be more specific, in this section, to take advantage of the superior capabilities of cosine function as a similarity metric and to address the issue with fine-tuning the hyperparameters of loss functions, we propose a novel hyperparameter-free objective function for deep metric learning, which incorporates N-pairs of data from multiple classes to measure the loss and optimize the network. The proposed objective function, which is referred to as the Multiple Pair Cosine Loss (MPCL), offers the following novelties:

(i) The MPCL does not impose any hyper-parameter fine-tuning step in the development of a DL model since the margin value, which indicates the separability of the class-specific embeddings is theoretically calculated in Subsection 5.4.3.

(ii) The MPCL provide more efficient generalization and less resource-exhaustive training over the input data, with potential applications in few-shot learning.

(iii) The MPCL automatically derives one-hot encoded embeddings for each class, while noticeably improving convergence speed compared to loss objectives based on softmax.

5.4.1 MPCL: Cosine-based Similarity Metric

As discussed previously, our proposed loss function is developed based on measuring the cosine similarity between the embeddings of input data and optimizing the network with the ultimate goal of maximizing the inter-class and minimizing the intra-class similarity between the embeddings. Due to utilization of cosine function as the similarity metric, we would ideally expect to obtain the similarity score of 1 for intra-class instances and 0 the inter-class ones. To formulate our argument, we consider the network architecture shown in Figure 5.7, which is designed for a N-class classification problem. Defining $\boldsymbol{X}$ as the tensor representing an instance of the input data, in each iteration of the training process, one instance from each class of the data is collected to form an input set, i.e., $\{\boldsymbol{X}, \boldsymbol{X}^{\mathscr{P}}, \boldsymbol{X}^{\mathscr{N}_1}, \ldots, \boldsymbol{X}^{\mathscr{N}_{N-1}}\}$, where $\boldsymbol{X}^{\mathscr{P}}$ represents a positive instance and $\boldsymbol{X}^{\mathscr{N}_1}, \ldots, \boldsymbol{X}^{\mathscr{N}_{N-1}}$ represent negative instances from the remaining $(N-1)$ classes. According to Figure 5.7, once the input data are passed through the encoding network, the output set of $\{\boldsymbol{o}^X, \boldsymbol{o}^{\mathscr{P}}, \boldsymbol{o}^1, \ldots, \boldsymbol{o}^{\mathscr{N}-1}\}$ is obtained, where the output of the network is assumed to

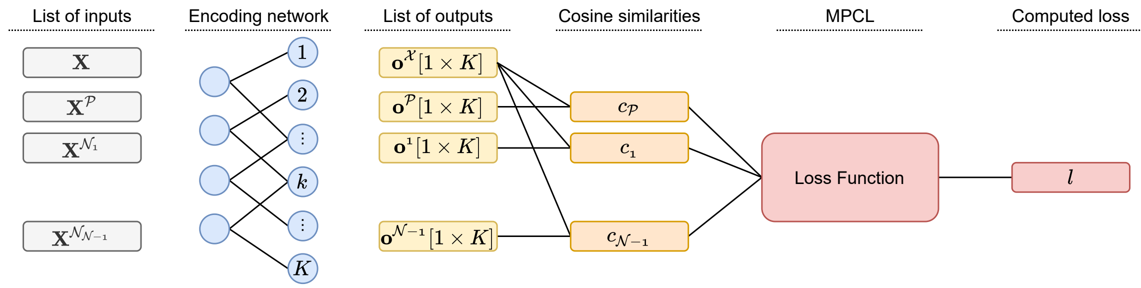


Figure 5.7: Architecture of a neural network based on the proposed objective function. Please note that in each iteration of the training process, $\{\boldsymbol{X}^{\mathscr{P}}, \boldsymbol{X}^{\mathscr{N}_1}, \ldots, \boldsymbol{X}^{\mathscr{N}_{N-1}}\}$ are fed to the encoding network and the objective function is applied on the collection of outputs.

be a vector of length K. Please note that for simplicity of the presentation, the batch size is assumed to be 1, thus the output of the encoding network is of size $(1 \times K)$.

Once the output set is obtained, the cosine similarity between $\boldsymbol{o}^X$ and any other output ($\boldsymbol{o}^y$) is calculated as

$$c_y = \frac{\boldsymbol{o}^X . \boldsymbol{o}^y}{|\boldsymbol{o}^X||\boldsymbol{o}^y|} = \frac{\sum_{k=1}^{K}(o_k^X o_k^y)}{\sqrt{\sum_{k=1}^{K}(o_k^X)^2}\sqrt{\sum_{k=1}^{K}(o_k^y)^2}}. \tag{5.4}$$

Upon calculation of $\{c_{\mathscr{P}}, c_1, \ldots, c_{\mathscr{N}-1}\}$, an objective function needs to be formulated such that $c_{\mathscr{P}} > c_1, \ldots, c_{\mathscr{N}-1}$. The proposed objective function, referred to as the MPCL is formulated as

$$l = \left(c_{\mathscr{P}} - \frac{1}{N-1}\sum_{j=1}^{N-1} c_j - \varepsilon\right)^2, \tag{5.5}$$

where ε defines a margin between the similarity of positive instances and the average similarity score of negative instances to prevent converging to trivial answers. The choice of ε identifies the degree of separation between the embeddings of different classes.

To better understand the proposed MPCL, Figure 5.8 depicts the value of loss with respect to the angle between positive instances $\theta_{\mathscr{P}}$ and the mean angle between the negative instances $\hat{\theta}_{\mathscr{N}}$, while setting $\varepsilon = 1$. Figure 5.8 (a) shows the loss space when $\theta_{\mathscr{P}}$ and $\hat{\theta}_{\mathscr{N}}$ can take any value in the range $[-\pi, \pi]$. As it can be observed in Figure 5.8 (a), Eq. (5.5) is an even function for both $\theta_{\mathscr{P}}$ and $\hat{\theta}_{\mathscr{N}}$, which affects fulfillment of the main purpose of this objective function (*consider the case that* $\theta_{\mathscr{P}} = 0$, $\hat{\theta}_{\mathscr{N}} = -\pi/2$, *and* $\varepsilon = 1$; *in this case also* $l = 0$ *but the network fails to converge to the right direction as fluctuating between* $\hat{\theta}_N = -\pi/2$ *to* $\hat{\theta}_N = +\pi/2$ *would not change the loss value*). Thus, we need to narrow down to a region in Figure 5.8 (a) where our proposed objective function is satisfied and the aforementioned issue is also avoided. By carefully inspecting the objective function and the loss space, it becomes evident that the range of $[0, \pi/2]$ for both $\theta_{\mathscr{P}}$ and $\hat{\theta}_{\mathscr{N}}$ points to the region of interest. This specific region, as shown in Figure 5.8 (b), implies that the network places the embeddings of different classes in perpendicular coordinates with respect to each other and all of the embeddings locate in a non-negative or non-positive closed orthant in $\mathbb{R}^K$. This property could be satisfied by proper selection of the activation function for the

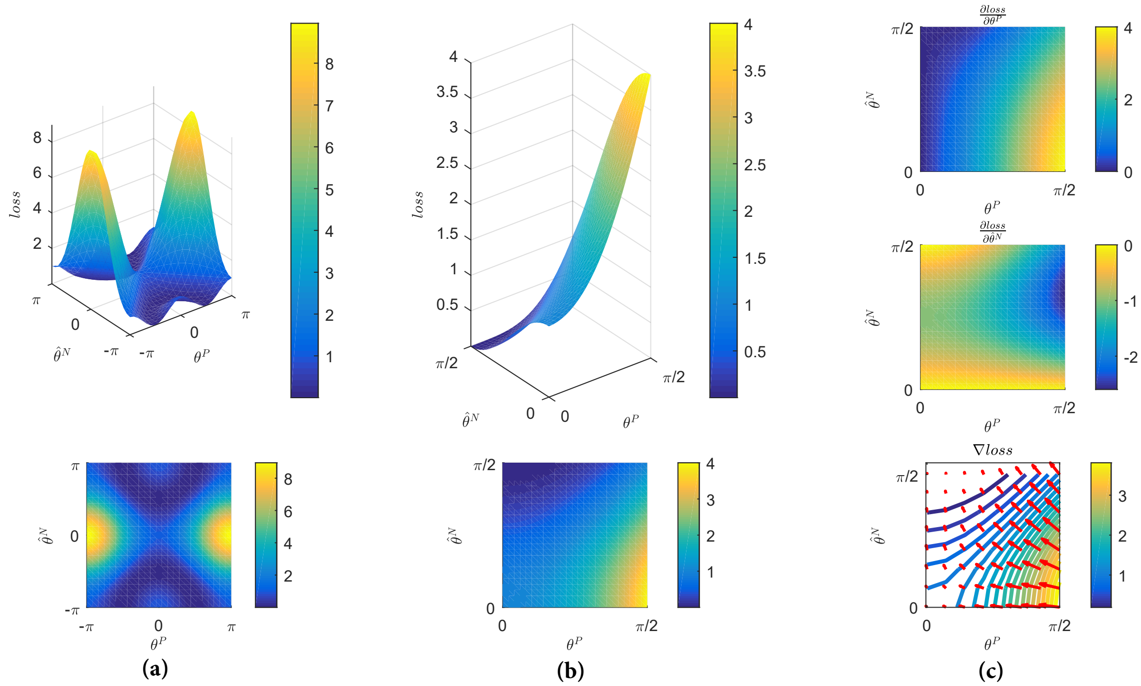


Figure 5.8: Simplified visualization of the loss space in MPCL. Please note that in these visualizations, to obtain a 3-dimensional sense of the loss space, $\hat{\theta}^N$ represents the space (in Radian) between $\boldsymbol{X}$ and all other negative instances. **(a)** The loss space when the embeddings are not forced to locate in the non-negative orthant of the $K-$dimensional space. **(b)** The loss space when embeddings are non-negative, which is satisfied by using non-negative activation functions in the last layer of the base neural network. **(c)** The magnitude of the gradient provided by the proposed objective function with respect to the separation between embeddings.

last layer of the encoding network. In other words, utilization of an activation function with outputs greater than zero ($\sigma(.) \geq 0$), e.g., ReLu or Sigmoid, suitably places all of the embeddings in the non-negative orthant of the $\mathbb{R}^K$ space.

5.4.2 Gradients in Back-propagation

In this subsection, we investigate the role that the proposed objective function plays in training and optimizing the base network by deriving its gradients in the backpropagation step. Please note that as the concept of backpropagation is thoroughly studied in the literature, here we solely focus on the gradients received by the last layer of the base network.

For simplicity of the presentation, we first derive the gradients for two output embeddings $\boldsymbol{o}^X$ and $\boldsymbol{o}^y$, and then generalize the results to the proposed objective function. Please note that i in what follows refers to the i^{th} element (node) of the last layer of the encoding network. Based on the chain rule, the gradient received by the last layer is

$\partial l/\partial \boldsymbol{o} = \partial l/\partial c \times \partial c/\partial \boldsymbol{o}$. Considering Eq. (5.4) we have

$$
\begin{aligned}
\frac{\partial c}{\partial o_i^X} &= \frac{o_i^y}{|\boldsymbol{o}^X||\boldsymbol{o}^y|} - \frac{o_i^X(\boldsymbol{o}^X.\boldsymbol{o}^y)}{|\boldsymbol{o}^X|^3|\boldsymbol{o}^y|} \\
\text{and} \quad \frac{\partial c}{\partial o_i^y} &= \frac{o_i^X}{|\boldsymbol{o}^X||\boldsymbol{o}^y|} - \frac{o_i^y(\boldsymbol{o}^X.\boldsymbol{o}^y)}{|\boldsymbol{o}^y|^3|\boldsymbol{o}^X|}.
\end{aligned} \tag{5.6}
$$

Considering Eq. (5.5), we get

$$
\begin{aligned}
\frac{\partial l}{\partial c_{\mathscr{P}}} &= 2\big(c_{\mathscr{P}} - \frac{1}{N-1}\sum_{j=1}^{N-1} c_j - \varepsilon\big) \\
\text{and} \quad \frac{\partial l}{\partial c_j} &= \frac{-2}{N-1}\big(c_{\mathscr{P}} - \frac{1}{N-1}\sum_{j=1}^{N-1} c_j - \varepsilon\big).
\end{aligned} \tag{5.7}
$$

Now, considering the network shown in Figure 5.7, the gradient received by Node k of the last layer of the base network (o_k) is given by

$$
\frac{\partial l}{\partial o_k} = \frac{\partial l}{\partial c_{\mathscr{P}}}\left(\frac{\partial c_{\mathscr{P}}}{\partial o_k^X} + \frac{\partial c_{\mathscr{P}}}{\partial o_k^{\mathscr{P}}}\right) + \frac{\partial l}{\partial c_j}\left(\frac{\partial c_1}{\partial o_k^X} + \frac{\partial c_1}{\partial o_k^1} + \ldots + \frac{\partial c_{N-1}}{\partial o_k^X} + \frac{\partial c_{N-1}}{\partial o_k^{N-1}}\right) \tag{5.8}
$$

By substituting Eqs. (5.6) and (5.7) in Eq. (5.8) we have

$$
\begin{aligned}
\frac{\partial l}{\partial o_k} =\, & 2\big(c_{\mathscr{P}} - \frac{1}{N-1}\sum_{j=1}^{N-1} c_j - \varepsilon\big)\left(\frac{o_k^{\mathscr{P}}}{|\boldsymbol{o}^X||\boldsymbol{o}^P|} - \frac{o_k^X(\boldsymbol{o}^X.\boldsymbol{o}^{\mathscr{P}})}{|\boldsymbol{o}^X|^3|\boldsymbol{o}^{\mathscr{P}}|} + \frac{o_k^X}{|\boldsymbol{o}^X||\boldsymbol{o}^{\mathscr{P}}|} - \frac{o_k^{\mathscr{P}}(\boldsymbol{o}^X.\boldsymbol{o}^{\mathscr{P}})}{|\boldsymbol{o}^P|^3|\boldsymbol{o}^X|}\right) + \\
& \frac{-2}{N-1}\big(c_{\mathscr{P}} - \frac{1}{N-1}\sum_{j=1}^{N-1} c_j - \varepsilon\big)\left(\sum_{j=1}^{N-1} \frac{o_k^j}{|\boldsymbol{o}^X||\boldsymbol{o}^j|} - \frac{o_k^X(\boldsymbol{o}^X.\boldsymbol{o}^j)}{|\boldsymbol{o}^X|^3|\boldsymbol{o}^j|} + \frac{o_k^X}{|\boldsymbol{o}^X||\boldsymbol{o}^j|} - \frac{o_k^j(\boldsymbol{o}^X.\boldsymbol{o}^j)}{|\boldsymbol{o}^j|^3|\boldsymbol{o}^X|}\right),
\end{aligned} \tag{5.9}
$$

which shows how the loss is backpropagated to the base network. This finalizes our discussion on the gradients of the MPCL.

5.4.3 Identification of Margin ε

In this subsection, we aim to identify a fixed value for the introduced margin (ε) in Eq. 5.5. Our approach is grounded on the fact that upon successful training of the network and achieving the desired objectives, we want the backpropagated gradient be equal to 0, which stops further updating the parameters of the network. To reiterate on the training objectives, it should be highlighted that our goal is to have $c_{\mathscr{P}} = 1$ and $\forall j; c_j = 0$ for all of the training samples, which means that inter-class embeddings are perfectly aligned with each other and the group of embeddings for each class are located in maximum distance with each other, i.e., $\theta = \pi/2$.

The case that $c_{\mathscr{P}} = cosinesimilarity(\boldsymbol{o}^X, \boldsymbol{o}^{\mathscr{P}}) = 1$ resembles the fact that $\boldsymbol{o}^X = m\boldsymbol{o}^{\mathscr{P}}$, where m is a positive number. Therefore, we have

$$
\begin{aligned}
& \frac{o_k^{\mathscr{P}}}{|\boldsymbol{o}^X||\boldsymbol{o}^{\mathscr{P}}|} - \frac{o_k^X(\boldsymbol{o}^X.\boldsymbol{o}^{\mathscr{P}})}{|\boldsymbol{o}^X|^3|\boldsymbol{o}^{\mathscr{P}}|} + \frac{o_k^X}{|\boldsymbol{o}^X||\boldsymbol{o}^{\mathscr{P}}|} - \frac{o_k^{\mathscr{P}}(\boldsymbol{o}^X.\boldsymbol{o}^{\mathscr{P}})}{|\boldsymbol{o}^{\mathscr{P}}|^3|\boldsymbol{o}^X|} \\
= \quad & \frac{o_k^{\mathscr{P}}}{m|\boldsymbol{o}^{\mathscr{P}}|^2} - \frac{o_k^{\mathscr{P}}}{m|\boldsymbol{o}^{\mathscr{P}}|^2} + \frac{o_k^{\mathscr{P}}}{|o^{\mathscr{P}}|^2} - \frac{o_k^{\mathscr{P}}}{|\boldsymbol{o}^{\mathscr{P}}|^2} \\
= \quad & 0.
\end{aligned}
\tag{5.10}
$$

By substituting the above information in Eq. (5.9) we get

$$
\frac{\partial l}{\partial o_i} = \frac{-2}{N-1}\big(1-\varepsilon\big)\Big(\sum_{j=1}^{N-1}\frac{o_i^j + o_i^X}{|\boldsymbol{o}^X||\boldsymbol{o}^j|}\Big). \tag{5.11}
$$

By capitalizing on the fact that embeddings are non-negative, it is clear that having $\forall j; o_i^j + o_i^X = 0$ would lead to $0/0$, which is impossible. Thus, to have $\frac{\partial l}{\partial o_i} = 0$ for the desired case, we need to set $\varepsilon = 1$. This relaxes the urge to fine-tune the margin as a hyper-parameter of the model, unlike the case for contrastive learning and triplet loss functions.

5.5 Simulation Results

To investigate the efficacy of the proposed MPCL in measuring the similarity of extracted emebeddings and serving as an objective function, we performed a thorough set of evaluations. Our evaluations are of three folds: (i) The case that negative instances from all of the classes are participated in calculating the loss in each update; (ii) The case that a few number of negative instances are participated in calculating the loss, and; (iii) The case that only a few number of training samples from each class are randomly selected to train the network. Please note that in all of the aforementioned evaluation scenarios, we have compared the results with the ones from triplet loss in Eq. (2.10) and MCNP [196] in Eq. (2.11). It is worth noting that in the literature of metric learning methods, to enhance generalization of the network and improve on classification accuracy, one common practice is employing data mining methods to form "good" training batches [252], which convey important information on (dis)similarity of classes. These methods include hard negative sampling, semi-hard negative sampling, batch all, and batch hard to name but a few. However, in this work none of the aforementioned techniques is employed as we want to particularly focus on the effect of the MPCL on optimizing the network to extract discriminative embeddings. Therefore, all the presented results in this section are obtained through random formation of the training and evaluation samples.

To fine-tune the hyper-parameters of the encoding network, we reserved 20% of the training data for validation purposes and our results revealed that network shown in Figure 5.9 yields promising accuracy for the three objective functions. Please note that as suggested by [196], L^2 regularization is applied to the output layer of the encoding network for the MCNP method.

To generate training batches, each training image is coupled with one positive instance

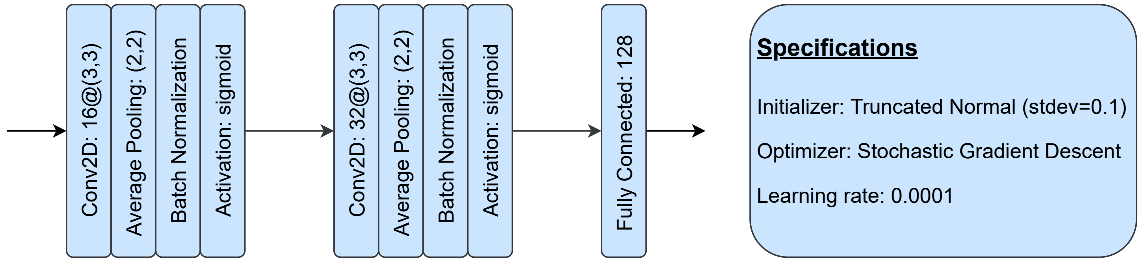


Figure 5.9: Architecture and the specifications of the encoding network in our experiments.

and $N-1$ number of negative instances, where all are selected randomly. Upon selection of each sample, it will be removed from the pool. This process is repeated for every training epoch. To evaluate the network over unseen images, we followed two strategies: (i) *Random Batch*, i.e., generating test batches following the same random procedure we explained for training batches, and; (ii) *Mean Embedding or Prototyping* [253], deriving the average embedding of each class and investigating the similarity of an unseen sample with each of the mean embedding vectors. Please note that in the former approach, each test sample is randomly coupled with positive and negative instances from training set.

- **Scenario 1:** The first approach to evaluate MPCL is when negative samples from all of the classes are participated in the training and testing phases. In this case, $54,200$ and $8,910$ number of training and testing batches are generated, respectively. As shown in Table 5.3 for $n = 9$, the MPCL outperforms its counterpart in both of the random batch and mean embedding approaches. To further investigate the performance of each method, the Figure 5.10 compares the classification accuracy of the three methods over the validation set and clearly demonstrates a faster convergence and higher accuracy for the MPCL. Please note that the plots in Figure 5.10 represent the “Random Batch” accuracy of the three methods, as they are obtained over the validation set.

- **Scenario 2:** The second set of evaluations investigate how the number of negative instances in the objective function change the final performance of the network. To this end, we evaluate the MPCL and MCNP for the numbers of $n =$ 1, 2, 5, and 7 negative samples. Please note that this experiment does not apply to triplet loss, as it is basically formulated based on one positive and one negative instance. The results of this experiment are tabulated in Table 5.3, where it is clearly observed that MPCL achieves better results and higher robustness for different cases of n.

- **Scenario 3:** In the last set of evaluations, we investigate the capability of the three objective functions in a “*few-shot learning*” scenario, which aims at learning the classification rule based on only a few number of training samples. We approached this problem by considering three cases of having $f =$ 5, 10, and 20 number of training samples for each class. The results of this test are presented in Table 5.4. Although MPCL fails to outperform the triplet loss in random batch testing scenario, it provides better classification accuracy in the mean embedding testing scenario.

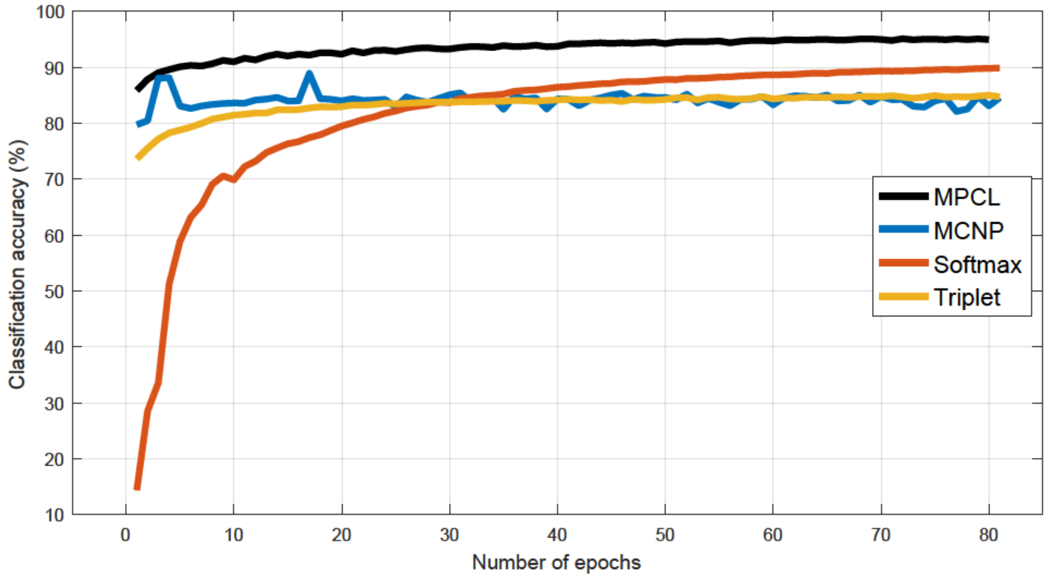


Figure 5.10: Comparison of classification accuracy for each method across the validation set. As the loss for each method follows a different scale, we used classification accuracy for our comparisons.

Table 5.3: The classification accuracy for different cases of n. Please note that n here denotes the number of negative instances participated in the objective function.

Method	Random Batch					Mean Embedding				
	$n=1$	$n=2$	$n=5$	$n=7$	$n=9$	$n=1$	$n=2$	$n=5$	$n=7$	$n=9$
MPCL	95.24	89.75	91.45	92.97	94.31	97.25	94.39	93.8	93.51	97.44
MCNP	93.48	87.24	74.3	67.32	86.21	77.93	77.94	78.29	78.26	88.91
Triplet	83	N/A	N/A	N/A	N/A	97.1	N/A	N/A	N/A	N/A

Table 5.4: The classification accuracy of the three methods in a few-shot learning scenario. Term f denotes the number of samples for each class.

Method	Random Batch			Mean Embedding		
	$f=5$	$f=10$	$f=20$	$f=5$	$f=10$	$f=20$
MPCL	62.26	68.17	75.87	68.52	71.69	81.29
MCNP	73.37	69.73	76.43	69.2	74.37	79.87
Triplet	83.36	84.96	83.73	64.52	71.27	76.76

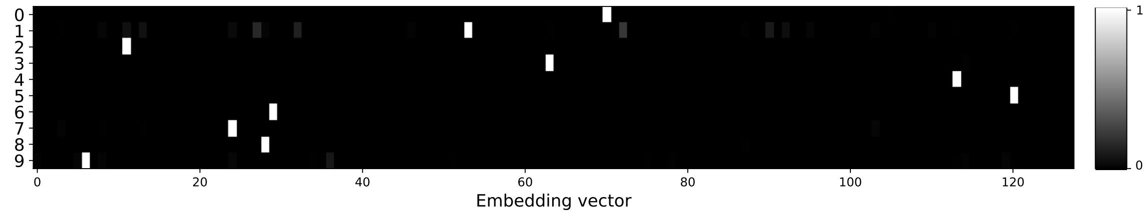


Figure 5.11: The class-specific average embeddings for MNIST dataset, obtained through MPCL.

5.5.1 Discussion

By having a closer look at Eq. (5.9), we can observe that the backpropagated gradients carry a mixture of the information obtained from positive and negative instances. This, collectively, adjusts the gradients in a way that our two desired objectives, i.e., maximum alignment for positive instances and maximum separation for negative instances, are satisfied. In other words, the ultimate goal of the proposed objective function is to extract class-specific embeddings that locate in the top-left corner of the loss space in Figure 5.8 (b). To better understand behavior of the MPCL in different regions of the loss space, it is worth investigating the magnitude of gradients with respect to $\theta_{\mathscr{P}}$ and $\hat{\theta}_{\mathscr{N}}$. As it is shown in Figure 5.8 (c)-top, MPCL reacts most when $\theta_{\mathscr{P}}$ is far from 0. We can observe high sensitivity to the value of $\theta_{\mathscr{P}}$ all over the region, except the parts that $\theta_{\mathscr{P}}$ is close to 0. In addition, Figure 5.8 (c)-middle shows the sensitivity of objective function where $0 < \hat{\theta}_{\mathscr{N}} < \pi/2$. For $\hat{\theta}_{\mathscr{N}} \to \pi/2$ the sensitivity is close to zero, which is desirable. However, sensitivity for $\hat{\theta}_{\mathscr{N}} \to 0$ is observed close to zero, which is not desirable, but this is compensated by the sensitivity of the objective function to $\theta_{\mathscr{P}}$ in that region. Figure 5.8 (c)-bottom depicts the gradients of the MPCL across the whole region and clearly shows how embeddings are pushed out of the bottom-right corner of the loss space to the top-left corner (from the worst case to the best case scenario).

As it is observed in the presented results from the three testing scenarios, the proposed MPCL method outperforms its counterparts. It is worth reiterating that as the scope of this work is investigating the features of a new objective function, we investigated all of the techniques in their basic format and avoided to employ any other technique to enhance the results. Commonly used techniques to orchestrate the training samples and enhance the performance and generalization of the network, e.g., hard negative sampling and semi-hard negative sampling, have led to better results on MCNP and Triplet losses, and we expect the same for the MPCL.

The fundamental idea of the MPCL is to extract embeddings from data in such a way that they span the non-negative orthant of feature space with maximum separability, i.e., orthogonality of class embeddings. Therefore, the MPCL can theoretically embed K classes of data with maximum separability in a K-dimensional feature space (K is the length of embedding vector). To validate this hypothesis, we have derived the average embedding vector for each class and the results are depicted in Figure 5.11. As it can be observed, the embeddings are in fact spars vectors with sparsity equal to $1/K$ and each class is embedded on one axis of the K-dimensional space. Please note that the difference between black and white pixels in Figure 5.11 is in the order of $[10e3 - 10e5]$, and only for visualization purposes, we have normalized the features of each class into the range of $[0-1]$.

It is worth highlighting that by employing one-hot encoded class labels and "*Softmax*" activation function for the output layer of the encoding network, we can achieve sparse embeddings similar to what we observe in Figure 5.11. However, in the MPCL no explicit instruction on this coding style is given to the network and this behavior is inferred by the network through fulfillment of the objective function. In addition, by revisiting the Figure 5.10 and comparing the convergence speed and classification accuracy for MPCL and the *Softmax* technique, we can clearly observe a faster convergence and higher classification accuracy over validation set for the MPCL method.

Broader Impact

The MPCL proposes a novel objective function for deep metric learning tasks based on cosine similarity of embeddings, which enhances the degree of generalization and the convergence speed. Deep metric learning has found increasing attention in classification, verification, and few-shot learning tasks, where limited number of training samples for each class are available. The MPCL optimizes the network based on the contrastive features of classes, rather than their characterizing features, and could be employed in various applications including but not limited to identity verification (e.g., face, and signature verification) for security purposes and diagnosis/classification of various diseases, where limited datasets restrain deployment of deep learning methods. That being said, the profound capacity of MPCL in requiring less training examples to generalize over the dissimilarities of studied domain could be adversely employed in crowd monitoring through surveillance cameras, which raises major disputes over the privacy of citizens. We believe that the undesirable and potentially nefarious applications of any machine learning technique, including MPCL, inevitably rise in parallel to their virtuous applications and the only robust way to control that aspect is by educating the practitioners in the AI domain on the ethical considerations of AI. In other words, although some ML techniques could be devised on an ad hoc basis to restrict or limit the efficacy of certain ML techniques in undesirable applications, there would always be a never ending chase to get around the restrictions, unless otherwise the ethical aspects of AI are incorporated in core courses of AI and sophisticated regulations are widely deployed.

5.6 Summary

In this chapter, the contribution of the thesis towards developing a DL-based screening protocol for the differential diagnosis of PD from ET was presented. Although the proposed NeurDNet framework is of paramount clinical importance to minimize the misdiagnosis rate of PD and ET, it can be effectively employed in various types of ADs to provide disease-specific assistive services. In other words, given the structural differences between the characteristics of PHT in PD and ET as discussed in Section 2.1.3, the differentiation of various disorders could potentially enable the AD to utilize a set of disease-specific parameters to maximize the quality of delivered services. This important contribution has not only provided an effective data-driven screening protocol for differentiation of PD and

ET with state-of-the-art classification accuracy but also provides clinically viable clues on the structural differences in PHT between PD and ET, which further facilitate the correct diagnosis of the two diseases.

Later on, we proposed the MPCL objective function for deep metric learning scenarios, which is more robust to local minimas and covariate shift, and relaxes the urge to employ regularizing or normalizing techniques in the network. MPCL incorporates negative instances from all of the classes to compute the loss and achieves a better generalization over the distribution of data. MPCL is compared with triplet loss and Multi-class N-pair objective loss in various scenarios over the MNIST dataset, and promising results were obtained. Moreover, the class-specific embeddings extracted through the MPCL objective function automatically converge to one-hot encoded labels of each class, while no instruction on this coding scheme is specifically given. We observed that the MPCL also outperforms a softmax-based classification scheme in terms of classification accuracy and speed of convergence.

Chapter 6

Contributions and Future Research Directions

6.1 Summary of Contributions

In the thesis, we presented our contributions towards the goal of devising innovative and effective data-driven processing frameworks for the development of rehabilitation and assistive technologies, which are specifically designed to assist individuals who are affected by age-related neurological disorders. To attain this objective, in general, one can introduce new communication mediums to the human brain to compensate for the lost functionalities of the patients, or develop certain technologies to retrieve the lost functionalities of the patients. In what follows, a summary of the contributions of the thesis towards fulfilling the first approach referred to as BCI systems, and the latter one referred to as AD, is presented.

1. **Development of ECCSP framework, which demonstrates the applicability of Error Correcting Output Coding (ECOC) classifiers for EEG studies [33]:** A BCI system designed to operate in real-world conditions, must be able to discriminate multiple tasks and activities. This fact expresses the urge to develop/implement classifiers intrinsically designed for multi-class problems. One such technique that is well regarded in other fields but has not yet been applied to EEG-based classification is the ECOC. The thesis fills the mentioned gap. The BCI Competition IV-2a dataset is used to evaluate the performance of the proposed ECCSP framework. Our results showed that ECCSP achieves similar performance in comparison to the state-of-the-art algorithms but is extensively simpler with significantly less computational overhead making it a practical alternative for real-time EEG motor imagery classification tasks.

2. **Development of ECCSP-TB framework, which introduces the Ternary-ECOC classification scheme, which boosts the classification performance in multiclass classification problems [34]:** In the thesis, a modified version of the ECOC classifiers is developed for EEG classification problems which deploys ternary class codewords. Therefore, more combinations of the classes and a greater number of classifiers vote for the final result. The proposed classifier is coupled with a Bayesian

framework to compute the optimized spatio-spectral filters to extract the most discriminative feature sets of different classes. The proposed framework is applied to a motor imagery classification problem and evaluated over the BCI Competition IV-2a dataset where the results indicate a noticeable enhancement over other methods developed for multi-class EEG classification.

3. **Development of ECCSP-TB2B framework which utilizes subject-specific optimized filter banks to extract informative features to analyze the MI signals [35, 36]:** To further individualize the spatial and spectral filter within an EEG processing framework in order to enhance the classification accuracy of the system, a novel Bayesian framework that simultaneously optimizes a number of subject-specific filter banks and spatial filters is developed. Optimized double-band spectro-spatial filters are derived based on common spatial patterns coupled with the ECOC classifiers. The proposed framework constructs optimized subject-specific spectral filters in an intuitive fashion resulting in the creation of significantly discriminant features, which is a crucial requirement for any EEG-based BCI system.

4. **Introducing a wavelet-based dimensionality-reduction scheme for EEG processing [37]:** With the goal of optimizing the EEG-based BCI system for real-time applications and reducing the processing workload while exploiting the maximum amount of information from the EEG signals, the thesis proposes a level-based classification approach that couples the Wavelet decomposition with Riemannian manifold spatial learning (WvRiem). In the proposed WvRiem framework, the EEG signals are decomposed into several components (levels) and then spatial filtering via Riemannian manifold learning is performed on the best level which yields the most discriminating features. The proposed WvRiem is evaluated on the BCI Competition IV-2a dataset and noticeably outperforms its counterparts.

5. **Development of a deep learning framework based on Siamese neural networks for EEG processing [39]:** Despite the successful employment of deep learning methods in various domains, their application for small medical datasets always raises concerns about the curse of overfitting to the training data. The thesis addresses this unmet quest by proposing a new EEG processing and feature extraction paradigm based on Siamese neural networks, which can be conveniently merged and scaled up for multi-class problems. The idea of Siamese networks is to train a double-input neural network based on a contrastive loss-function, which provides the capability of verifying if two input EEG trials are from the same class or not. The introduced Siamese architecture, which is developed based on Convolutional Neural Networks (CNN) and provides a binary output on the similarity of two inputs, is combined with One vs. Rest (OVR) and One vs. One (OVO) techniques to scale up for multi-class problems. The efficacy of this architecture is evaluated on a 4-class Motor Imagery (MI) dataset from BCI Competition IV-2a and the results suggest a promising performance compared to its counterparts.

6. **Development of an adaptive processing framework based on wavelet transformation and Kalman filtering to predict the voluntary component of hand motion real-time and in a myopic fashion [40,41]:** One major bottleneck in accurately separating the voluntary and involuntary components of hand motion in patients with PHT is the highly diverse and dynamic behavioral pattern of PHT in and across patients. The thesis addresses this issue by developing a novel on-line adaptive method which can adjust the hyper-parameters of the filter to the variable characteristics of the tremor. The proposed Wavelet decomposition coupled with adaptive Kalman filtering technique for pathological tremor Extraction, referred to as the WAKE framework, is composed of a new adaptive Kalman filter and a wavelet transform core to provide an indirect prediction of the tremor, one sample ahead of time, to be used for its suppression. The performance of WAKE is evaluated over three different datasets, where the first one is a synthetic dataset that simulates hand tremor under ten different conditions. The second and third ones are real datasets recorded from patients with PHT. The results obtained from the proposed WAKE framework demonstrate significant improvements in the estimation accuracy in comparison with two well-regarded techniques in the literature.

7. **Development of a deep learning framework based on recurrent neural networks to discriminate the voluntary and involuntary components of motion in real-time and in a myopic fashion [42, 44]:** Another major issue with estimating the voluntary and involuntary components of hand motion in patients with PHT is the unavailability of ground truth to precisely validate different processing frameworks developed for this task. The thesis addresses this unmet need by establishing a deep recurrent model to predict and eliminate the PHT component of hand motion. More specifically, we propose a machine learning-based, assumption-free, and real-time PHT elimination framework, the PHTNet, by incorporating deep bidirectional recurrent neural networks. The PHTNet is developed over a hand motion dataset of 81 ET and PD patients collected systematically in a movement disorders clinic over 3 years. The PHTNet is the first intelligent systems model developed on this scale for PHT elimination that maximizes the resolution of estimation and allows for the prediction of future and upcoming sub-movements.

8. **Performing a feasibility study on fusing two different datasets on hand motion recordings to train a neural network [43]:** Despite the successful employment of deep learning methods in various domains, their application for small medical datasets always raises concerns about the curse of overfitting to the training data. Since the availability of large datasets, especially in the PHT estimation field is a bottleneck, the thesis investigates the possibility of combining different recording modalities of PHT to generate a neural network for this purpose. In fact, the thesis approves the possibility of jointly using accelerometer data and gyroscope recordings to produce a larger dataset for training a relatively complex network, which can potentially be extended for a deeper generalization.

9. **Development of a Data-driven Method for Discrimination of Neurological Disorders [45]:** Societal aging has drastically increased the prevalence of age-related

neurological disorders worldwide, such as Parkinson's Disease (PD) and Essential Tremor (ET). Pathological Hand Tremor (PHT) is a common symptom of PD and ET, which affects manual targeting, motor coordination, and movement kinetics. Effective treatment and management of the symptoms rely on the correct and in-time diagnosis of the affected individuals, where the characteristics of PHT serve as an imperative metric for this purpose. Due to the overlapping features of the corresponding symptoms, however, a high level of expertise and specialized diagnostic methodologies are required to correctly distinguish PD from ET. The thesis proposes a data-driven model, referred to as NeurDNet, which processes the kinematics of the hand in the affected individuals and classifies the patients into PD or ET. NeurDNet is trained over 90 hours of hand motion signals consisting of 250 tremor assessments from 81 patients, recorded at the London Movement Disorders Centre, ON, Canada. The NeurDNet outperforms its state-of-the-art counterparts achieving exceptional differential diagnosis accuracy of 95.55%. In addition, using the explainability and interpretability measures for machine learning models, clinically viable and statistically significant insights on how the data-driven model discriminates between the two groups of patients are achieved.

10. **Cosine-based Objective Function for Deep Metric Learning [46]:** The unprecedented capacity of deep learning techniques in extracting high-level semantic embeddings from data has catalyzed the potent of deep metric learning for classification, verification, few-shot learning, and visual search tasks. A major bottleneck in further boosting the performance of deep metric learning is the objective function employed for recognizing the distance (similarity) of embeddings. Existing methods e.g., contrastive loss and triplet loss, often suffer from slow convergence and poor local minimum due to the utilization of only one negative instance to generalize over the pairwise distance between data points. In addition, some existing methods are solely based on the Euclidean distance between the embeddings, which makes the model significantly sensitive to the covariate shift. The thesis proposes a multi-class N-pair objective function based on cosine similarity, referred to as MPCL, where the intra-class and inter-class alignment of group embeddings are respectively maximized and minimized. Upon successful training of the network, the embeddings of each class find perpendicular directions with respect to each other, spanning in the non-negative orthant of the features space. The MPCL is thoroughly evaluated over the MNIST dataset and promising results in terms of classification accuracy and speed of convergence are obtained.

6.2 Future Research Directions

As it is highlighted throughout the thesis, the ultimate goal of the research works presented here is to develop effective and reliable processing frameworks to be employed in various types of ADs. The contributions of the thesis can independently serve as stand-alone processing units in different applications or they can be aggregated and integrated into an AD, e.g., the application depicted in Figure 1.1, to enhance the overall performance of the

AD. In continuation of the research works presented in the thesis, the following potential directions could be followed.

1. Given the sample AD shown in Figure 1.1, one promising direction is to develop efficient methods for the "Sensor fusion and decision making" block for different applications such as FES systems. In fact, this block determines how and when the information from the BCI block and the PHT processing unit should be aggregated in order to deliver the expected assistive service. The two blocks could be arranged to work in parallel with each other so that the communication bandwidth increases and a wider range of disabilities can be addressed. Otherwise, they can form a sequential architecture with the ultimate goal of enhancing the reliability and performance of the system. In fact, the latter solution offers an architecture analogous to a well-regarded ML technique, referred to as a mixture of experts.

2. Another promising direction is to collect and analyze an inclusive dataset of limb movements in patients with neurological disorders by means of commercial wearable devices, e.g. smartwatch. This potential direction relaxes the urge to employ medical-grade devices for screening and diagnosis of the patients, enabling the development of cost-effective and commercializable assistive devices. Along with this topic, a comprehensive study on the explainability of neural models trained for diagnostic purposes is an important topic to follow.

3. Regarding the part on discriminating the voluntary and involuntary components of hand motion signals, considering a sophisticated dynamic model of hand motion, which incorporates the effect of different muscles in forming a movement, would enhance the accuracy of modeling and estimation. Moreover, this modeling scheme could precisely identify the tremorous muscle to be targeted by an FES system.

4. Regarding the direction on MI-based BCI, Riemannian manifold learning due to its capacity in spatial filtering and source localization of EEG signals yields promising results in BCI applications, which can be effectively coupled with the magnificent learning capacity of neural networks and enhance the accuracy of classification.

5. In the thesis, to estimate the voluntary and involuntary components of hand motion, RNNs were employed to process the hand motion signals. An alternative to RNN is the employment of Variational Auto Encoders (VAE) to process the signal and estimate either the voluntary or involuntary component of hand motion. In particular, Denoising VAEs are the prime choice for such types of applications, where the goal is to estimate a signal (voluntary component) from a noise-contaminated measurement.

6. The ECCSP-TB2B framework, which proposes an optimization process based on Bayesian methods to derive subject-specific spectral and spatial filters could be modified such that the optimization process is satisfied through Stochastic Gradient Descent (SGD) method. Technically, SGD is widely employed for training neural networks and offers a less computationally expensive solution for optimizing the spectral filters in the EEG processing pipeline.

7. As a promising direction, the integration of NeurDNet in an AD to provide disease-specific assistive services could be studied. To be more specific, investigating the effect of disease on the parameters and hyper-parameters of the BCI module and/or the PHT processing framework serves as the main building block for this study.

Bibliography

[1] S. N. Abdulkader, A. Atia, and M.-S. M. Mostafa, "Brain computer interfacing: Applications and challenges," *Egyptian Informatics Journal*, vol. 16, no. 2, pp. 213–230, 2015.

[2] U. Nations, *World population prospects, the 2017 Revision, Volume I: comprehensive tables*. Department of Economics and Social Affairs, Population Devision, 2017.

[3] M. J. Daley and W. L. Spinks, "Exercise, mobility and aging," *Sports medicine*, vol. 29, no. 1, pp. 1–12, 2000.

[4] A. B. Schultz, "Mobility impairment in the elderly: challenges for biomechanics research," *Journal of biomechanics*, vol. 25, no. 5, pp. 519–528, 1992.

[5] P. E. Martin and M. D. Grabiner, "Aging, exercise, and the predisposition to falling," *Journal of applied biomechanics*, vol. 15, no. 1, pp. 52–55, 1999.

[6] C. Kotsavasiloglou, N. Kostikis, D. Hristu-Varsakelis, and M. Arnaoutoglou, "Machine learning-based classification of simple drawing movements in parkinson's disease," *Biomedical Signal Processing and Control*, vol. 31, pp. 174–180, 2017.

[7] C. Camara, P. Isasi, K. Warwick, V. Ruiz, T. Aziz, J. Stein, and E. Bakštein, "Resting tremor classification and detection in parkinson's disease patients," *Biomedical Signal Processing and Control*, vol. 16, pp. 88–97, 2015.

[8] E. J. Smits, A. J. Tolonen, L. Cluitmans, M. Van Gils, R. C. Zietsma, R. W. Borgemeester, T. van Laar, and N. M. Maurits, "Graphical tasks to measure upper limb function in patients with parkinson's disease: Validity and response to dopaminergic medication," *IEEE journal of biomedical and health informatics*, vol. 21, no. 1, pp. 283–289, 2017.

[9] J. I. Serrano, J. P. Romero, M. D. del Castillo, E. Rocon, E. D. Louis, and J. Benito-León, "A data mining approach using cortical thickness for diagnosis and characterization of essential tremor," *Scientific Reports*, vol. 7, no. 1, p. 2190, dec 2017. [Online]. Available: http://www.nature.com/articles/s41598-017-02122-3

[10] R. Pahwa and K. E. Lyons, *Handbook of Essential Tremor and Other Tremor Disorders*. CRC Press, 2005.

[11] C. Liu, J. Wang, B. Deng, X. Wei, H. Yu, H. Li, C. Fietkiewicz, and K. A. Loparo, "Closed-loop control of tremor-predominant parkinsonian state based on parameter estimation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 10, pp. 1109–1121, 2016.

[12] L. Ferrucci, R. Cooper, M. Shardell, E. M. Simonsick, J. A. Schrack, and D. Kuh, "Age-related change in mobility: perspectives from life course epidemiology and geroscience," *Journals of Gerontology Series A: Biomedical Sciences and Medical Sciences*, vol. 71, no. 9, pp. 1184–1194, 2016.

[13] G. R. Müller-Putz, C. Breitwieser, F. Cincotti, R. Leeb, M. Schreuder, F. Leotta, M. Tavella, L. Bianchi, A. Kreilinger, A. Ramsay *et al.*, "Tools for brain-computer interaction: a general concept for a hybrid bci," *Frontiers in neuroinformatics*, vol. 5, p. 30, 2011.

[14] S. Makeig, C. Kothe, T. Mullen, N. Bigdely-Shamlo, Z. Zhang, and K. Kreutz-Delgado, "Evolving signal processing for brain–computer interfaces," *Proceedings of the IEEE*, vol. 100, no. Special Centennial Issue, pp. 1567–1584, 2012.

[15] B. A. Osuagwu, M. Zych, and A. Vuckovic, "Is implicit motor imagery a reliable strategy for a brain–computer interface?" *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 12, pp. 2239–2248, 2017.

[16] P. Aricò, G. Borghini, G. Di Flumeri, N. Sciaraffa, A. Colosimo, and F. Babiloni, "Passive bci in operational environments: insights, recent advances, and future trends," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 7, pp. 1431–1436, 2017.

[17] A. S. Aghaei, B. Donmez, C. C. Liu, D. He, G. Liu, K. N. Plataniotis, H.-Y. W. Chen, and Z. Sojoudi, "Smart driver monitoring: when signal processing meets human factors: in the driver's seat," *IEEE Signal Processing Magazine*, vol. 33, no. 6, pp. 35–48, 2016.

[18] G. Schirner, D. Erdogmus, K. Chowdhury, and T. Padir, "The future of human-in-the-loop cyber-physical systems," *Computer*, vol. 46, no. 1, pp. 36–45, 2013.

[19] J. J. Daly and J. R. Wolpaw, "Brain–computer interfaces in neurological rehabilitation," *The Lancet Neurology*, vol. 7, no. 11, pp. 1032–1043, 2008.

[20] R. Xu, N. Jiang, N. Mrachacz-Kersting, C. Lin, G. A. Prieto, J. C. Moreno, J. L. Pons, K. Dremstrup, and D. Farina, "A closed-loop brain–computer interface triggering an active ankle–foot orthosis for inducing cortical neural plasticity," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 7, pp. 2092–2101, 2014.

[21] M. Grosse-Wentrup, D. Mattia, and K. Oweiss, "Using brain–computer interfaces to induce neural plasticity and restore function," *Journal of neural engineering*, vol. 8, no. 2, pp. 1–5, 2011.

[22] R. Leeb, L. Tonin, M. Rohm, L. Desideri, T. Carlson, and J. d. R. Millan, "Towards independence: a bci telepresence robot for people with severe motor disabilities," *Proceedings of the IEEE*, vol. 103, no. 6, pp. 969–982, 2015.

[23] L. Van Dokkum, T. Ward, and I. Laffont, "Brain computer interfaces for neurorehabilitation–its current status as a rehabilitation strategy post-stroke," *Annals of physical and rehabilitation medicine*, vol. 58, no. 1, pp. 3–8, 2015.

[24] O. Talakoub, C. Marquez-Chin, M. R. Popovic, J. Navarro, E. T. Fonoff, C. Hamani, and W. Wong, "Reconstruction of reaching movement trajectories using electrocorticographic signals in humans," *PloS one*, vol. 12, no. 9, pp. 1–19, 2017.

[25] H. Yuan and B. He, "Brain–computer interfaces using sensorimotor rhythms: current state and future perspectives," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 5, pp. 1425–1435, 2014.

[26] A. Anouti and W. C. Koller, "Tremor disorders; diagnosis and management," *Western journal of medicine*, vol. 162, no. 6, pp. 510–513, 1995.

[27] G. Grimaldi and M. Manto, "Tremor: from pathogenesis to treatment," *Synthesis lectures on biomedical engineering*, vol. 3, no. 1, pp. 1–212, 2008.

[28] A. Benazzouz and M. Hallett, "Mechanism of action of deep brain stimulation," *Neurology*, vol. 55, no. 12 Suppl 6, 2000.

[29] E. Rocon, M. Manto, J. Pons, S. Camut, and J. M. Belda, "Mechanical suppression of essential tremor," *The Cerebellum*, vol. 6, no. 1, pp. 73–78, 2007.

[30] E. Rocon, J. Belda-Lois, A. Ruiz, M. Manto, J. C. Moreno, and J. L. Pons, "Design and validation of a rehabilitation robotic exoskeleton for tremor assessment and suppression," *IEEE Transactions on neural systems and rehabilitation engineering*, vol. 15, no. 3, pp. 367–378, 2007.

[31] M. Javidan, J. Elek, and A. Prochazka, "Attenuation of pathological tremors by functional electrical stimulation ii: clinical evaluation," *Annals of biomedical engineering*, vol. 20, no. 2, pp. 225–236, 1992.

[32] A. Prochazka, J. Elek, and M. Javidan, "Attenuation of pathological tremors by functional electrical stimulation i: Method," *Annals of biomedical engineering*, vol. 20, no. 2, pp. 205–224, 1992.

[33] S. Shahtalebi and A. Mohammadi, "Error correction output codding coupled with the csp for motor imagery bci systems," in *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 2071–2075.

[34] ——, "Ternary ecoc classifiers coupled with optimized spatio-spectral patterns for multiclass motor imagery classification," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2017, pp. 2231–2236.

[35] ——, “A bayesian framework to optimize double band spectra spatial filters for motor imagery classification,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 871–875.

[36] ——, “Bayesian optimized spectral filters coupled with ternary ecoc for single-trial eeg classification,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 12, pp. 2249–2259, 2018.

[37] ——, “Feature space reduction for single trial eeg classification based on wavelet decomposition,” in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2019, pp. 7161–7164.

[38] M. Mirgholami, S. Shahtalebi, W. Cui, R. Karimi, A. Asif, and A. Mohammadi, “Adaptive subject-specific bayesian spectral filtering for single trial eeg classification,” in *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2019, pp. 1–5.

[39] S. Shahtalebi, A. Asif, and A. Mohammadi, “Siamese neural networks for eeg-based brain-computer interfaces,” in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, 2020, pp. 442–446.

[40] S. Shahtalebi, A. Mohammadi, S. F. Atashzar, and R. V. Patel, “A multi-rate and auto-adjustable wavelet decomposition framework for pathological hand tremor extraction,” in *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2017, pp. 432–436.

[41] S. Shahtalebi, S. F. Atashzar, R. V. Patel, and A. Mohammadi, “Wake: Wavelet decomposition coupled with adaptive kalman filtering for pathological tremor extraction,” *Biomedical Signal Processing and Control*, vol. 48, pp. 179–188, 2019.

[42] ——, “Hmfp-dbrnn: Real-time hand motion filtering and prediction via deep bidirectional rnn,” *IEEE Robotics and Automation Letters*, 2019.

[43] ——, “Training of deep bidirectional rnns for hand motion filtering via multimodal data fusion.” in *GlobalSIP*, 2019, pp. 1–5.

[44] S. Shahtalebi, S. F. Atashzar, O. Samotus, R. V. Patel, M. S. Jog, and A. Mohammadi, “Phtnet: characterization and deep mining of involuntary pathological hand tremor using recurrent neural network models,” *Scientific reports*, vol. 10, no. 1, pp. 1–19, 2020.

[45] S. Shahtalebi, S. F. Atashzar, R. V. Patel, M. S. Jog, and A. Mohammadi, “Neurdnet: An artificial intelligent-based approach in neurological disorders classification,” *Submitted to Scientific reports*, 2020.

[46] S. Shahtalebi and A. Mohammadi, “Mpcl: Deep metric learning based on multiclass n-pair cosine similarity,” *Submitted to IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[47] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain–computer interfaces for communication and control," *Clinical neurophysiology*, vol. 113, no. 6, pp. 767–791, 2002.

[48] B. Z. Allison, E. W. Wolpaw, and J. R. Wolpaw, "Brain–computer interface systems: progress and prospects," *Expert review of medical devices*, vol. 4, no. 4, pp. 463–474, 2007.

[49] R. Leeb, H. Sagha, R. Chavarriaga, and J. del R Millán, "A hybrid brain–computer interface based on the fusion of electroencephalographic and electromyographic activities," *Journal of neural engineering*, vol. 8, no. 2, pp. 1–5, 2011.

[50] J. d. R. Millán, R. Rupp, G. Müller-Putz, R. Murray-Smith, C. Giugliemma, M. Tangermann, C. Vidaurre, F. Cincotti, A. Kubler, R. Leeb *et al.*, "Combining brain–computer interfaces and assistive technologies: state-of-the-art and challenges," *Frontiers in neuroscience*, vol. 4, no. 161, pp. 1–16, 2010.

[51] G. Pfurtscheller, B. Z. Allison, G. Bauernfeind, C. Brunner, T. Solis Escalante, R. Scherer, T. O. Zander, G. Mueller-Putz, C. Neuper, and N. Birbaumer, "The hybrid bci," *Frontiers in neuroscience*, vol. 4, no. 30, pp. 1–11, 2010.

[52] H. Ramoser, J. Muller-Gerking, and G. Pfurtscheller, "Optimal spatial filtering of single trial eeg during imagined hand movement," *IEEE transactions on rehabilitation engineering*, vol. 8, no. 4, pp. 441–446, 2000.

[53] X. Xie, Z. L. Yu, H. Lu, Z. Gu, and Y. Li, "Motor imagery classification based on bilinear sub-manifold learning of symmetric positive-definite matrices," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 6, pp. 504–516, 2016.

[54] K. K. Ang, Z. Y. Chin, C. Wang, C. Guan, and H. Zhang, "Filter bank common spatial pattern algorithm on bci competition iv datasets 2a and 2b," *Frontiers in neuroscience*, vol. 6, no. 39, pp. 1–9, 2012.

[55] H. Lu, H.-L. Eng, C. Guan, K. N. Plataniotis, and A. N. Venetsanopoulos, "Regularized common spatial pattern with aggregation for eeg classification in small-sample setting," *IEEE transactions on Biomedical Engineering*, vol. 57, no. 12, pp. 2936–2946, 2010.

[56] F. Lotte and C. Guan, "Regularizing common spatial patterns to improve bci designs: unified theory and new algorithms," *IEEE Transactions on biomedical Engineering*, vol. 58, no. 2, pp. 355–362, 2011.

[57] A. S. Aghaei, M. S. Mahanta, and K. N. Plataniotis, "Separable common spatio-spectral patterns for motor imagery bci systems," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 1, pp. 15–29, 2016.

[58] X. Li, C. Guan, H. Zhang, and K. K. Ang, “A unified fisher’s ratio learning method for spatial filter optimization,” *IEEE transactions on neural networks and learning systems*, vol. 28, no. 11, pp. 2727–2737, 2017.

[59] H.-I. Suk and S.-W. Lee, “A novel bayesian framework for discriminative feature extraction in brain-computer interfaces,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 286–299, 2013.

[60] J. Shin, J. Kwon, and C.-H. Im, “A ternary hybrid eeg-nirs brain-computer interface for the classification of brain activation patterns during mental arithmetic, motor imagery, and idle state,” *Frontiers in neuroinformatics*, vol. 12, no. 5, pp. 1–9, 2018.

[61] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[62] M. Lin, Q. Chen, and S. Yan, “Network in network,” *International Conference on Learning Representations (ICLR)*, 2014.

[63] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[64] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” in *European conference on computer vision*. Springer, 2016, pp. 630–645.

[65] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and accurate deep network learning by exponential linear units (elus),” *International Conference on Learning representations (ICLR)*, 2015.

[66] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[67] T. Salimans and D. P. Kingma, “Weight normalization: A simple reparameterization to accelerate training of deep neural networks,” in *Advances in Neural Information Processing Systems*, 2016, pp. 901–909.

[68] S. Sakhavi, C. Guan, and S. Yan, “Learning temporal information for brain-computer interface using convolutional neural networks,” *IEEE transactions on neural networks and learning systems*, no. 99, pp. 1–11, 2018.

[69] H. Cecotti, M. P. Eckstein, and B. Giesbrecht, “Single-trial classification of event-related potentials in rapid serial visual presentation tasks using supervised spatial filtering,” *IEEE transactions on neural networks and learning systems*, vol. 25, no. 11, pp. 2030–2042, 2014.

[70] S. Stober, D. J. Cameron, and J. A. Grahn, "Using convolutional neural networks to recognize rhythm stimuli from electroencephalography recordings," in *Advances in neural information processing systems*, 2014, pp. 1449–1457.

[71] S. Stober, A. Sternin, A. M. Owen, and J. A. Grahn, "Deep feature learning for eeg recordings," *arXiv preprint arXiv:1511.04306*, 2015.

[72] "Eeg-based prediction of driver's cognitive performance by deep convolutional neural network," *Signal Processing: Image Communication*, vol. 47, pp. 549 – 555, 2016.

[73] S. Sakhavi, C. Guan, and S. Yan, "Parallel convolutional-linear neural network for motor imagery classification," in *2015 23rd European Signal Processing Conference (EUSIPCO)*. IEEE, 2015, pp. 2736–2740.

[74] O. Dick and A. Nozdrachev, "Features of parkinsonian and essential tremor of the human hand," *Human Physiology*, vol. 42, no. 3, pp. 271–278, 2016.

[75] D. Tarsy, "Handbook of essential tremor and other tremor disorders," *Neurology*, vol. 70, no. 4, pp. 326–326, 2008. [Online]. Available: https://n.neurology.org/content/70/4/326.1

[76] S. Tatinati, K. C. Veluvolu, and W. T. Ang, "Multistep prediction of physiological tremor based on machine learning for robotics assisted microsurgery," *IEEE Transactions on Cybernetics*, vol. 45, no. 2, pp. 328–339, 2015.

[77] C. N. Riviere, W. T. Ang, and P. K. Khosla, "Toward active tremor canceling in handheld microsurgical instruments," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 5, pp. 793–800, 2003.

[78] S. Wang, Y. Gao, J. Zhao, and H. Cai, "Adaptive sliding bandlimited multiple fourier linear combiner for estimation of pathological tremor," *Biomedical Signal Processing and Control*, vol. 10, pp. 260–274, 2014.

[79] R. Bhidayasiri, "Differential diagnosis of common tremor syndromes," *Postgraduate Medical Journal*, vol. 81, no. 962, pp. 756–762, 2005.

[80] G. Deuschl, J. Raethjen, M. Lindemann, and P. Krack, "The pathophysiology of tremor," *Muscle & nerve*, vol. 24, no. 6, pp. 716–735, 2001.

[81] S. F. Atashzar, M. Shahbazi, O. Samotus, M. Tavakoli, M. S. Jog, and R. V. Patel, "Characterization of upper-limb pathological tremors: application to design of an augmented haptic rehabilitation system," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 5, pp. 888–903, Aug 2016.

[82] A. Hossen, M. Muthuraman, J. Raethjen, G. Deuschl, and U. Heute, "Discrimination of parkinsonian tremor from essential tremor by implementation of a wavelet-based soft-decision technique on emg and accelerometer signals," *Biomedical Signal Processing and Control*, vol. 5, no. 3, pp. 181–188, 2010.

[83] F. Xiao, Y. Gao, S. Wang, and J. Zhao, "Prediction of pathological tremor using adaptive multiple oscillators linear combiner," *Biomedical Signal Processing and Control*, vol. 27, pp. 77–86, 2016.

[84] B. Taheri, D. Case, and E. Richer, "Adaptive suppression of severe pathological tremor by torque estimation method," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 2, pp. 717–727, 2015.

[85] K. C. Veluvolu, S. Tatinati, S.-M. Hong, and W. T. Ang, "Multistep prediction of physiological tremor for surgical robotics applications," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 11, pp. 3074–3082, 2013.

[86] J. P. Giuffrida, D. E. Riley, B. N. Maddux, and D. A. Heldman, "Clinically deployable kinesia™ technology for automated tremor assessment," *Movement disorders: official journal of the Movement Disorder Society*, vol. 24, no. 5, pp. 723–730, 2009.

[87] A. Kucukelbir, A. Kushki, and K. N. Plataniotis, "A new stochastic estimator for tremor frequency tracking," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2009, pp. 421–424.

[88] S. Dosen, S. Muceli, J. L. Dideriksen, J. P. Romero, E. Rocon, J. Pons, and D. Farina, "Online tremor suppression using electromyography and low-level electrical stimulation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 3, pp. 385–395, 2015.

[89] K. Kiguchi and Y. Hayashi, "An emg-based control for an upper-limb power-assist exoskeleton robot," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 4, pp. 1064–1071, 2012.

[90] A. P. L. Bo, P. Poignet, and C. Geny, "Pathological tremor and voluntary motion modeling and online estimation for active compensation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 19, no. 2, pp. 177–185, 2011.

[91] K. C. Veluvolu and W. T. Ang, "Estimation of physiological tremor from accelerometers for real-time applications," *Sensors*, vol. 11, no. 3, pp. 3020–3036, 2011.

[92] S. Wang, Y. Gao, J. Zhao, and H. Cai, "Adaptive sliding bandlimited multiple fourier linear combiner for estimation of pathological tremor," *Biomedical Signal Processing and Control*, vol. 10, pp. 260–274, 2014.

[93] K. C. Veluvolu, W. T. Latt, and W. T. Ang, "Double adaptive bandlimited multiple fourier linear combiner for real-time estimation/filtering of physiological tremor," *Biomedical Signal Processing and Control*, vol. 5, no. 1, pp. 37–44, 2010.

[94] S. Tatinati, K. Nazarpour, W. T. Ang, and K. C. Veluvolu, "Multidimensional modeling of physiological tremor for active compensation in handheld surgical robotics," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 2, pp. 1645–1655, 2017.

[95] W. T. Latt, U.-X. Tan, C. Y. Shee, C. N. Riviere, and W. T. Ang, “Compact sensing design of a handheld active tremor compensation instrument,” *IEEE sensors journal*, vol. 9, no. 12, pp. 1864–1871, 2009.

[96] G. Deuschl, P. Bain, M. Brin, and A. H. S. Committee, “Consensus statement of the movement disorder society on tremor,” *Movement disorders*, vol. 13, no. S3, pp. 2–23, 1998.

[97] G. Deuschl, P. Krack, M. Lauk, and J. Timmer, “Clinical neurophysiology of tremor,” *Journal of Clinical Neurophysiology*, vol. 13, no. 2, pp. 110–121, 1996.

[98] B. Molparia, B. N. Schrader, E. Cohen, J. L. Wagner, S. R. Gupta, S. Gould, N. Hwynn, E. G. Spencer, and A. Torkamani, “Combined accelerometer and genetic analysis to differentiate essential tremor from parkinson’s disease,” *PeerJ*, vol. 6, pp. 1–15, 2018.

[99] D. M. Gillard, T. Cameron, A. Prochazka, and M. J. Gauthier, “Tremor suppression using functional electrical stimulation: A comparison between digital and analog controllers,” *IEEE transactions on Rehabilitation Engineering*, vol. 7, no. 3, pp. 385–388, 1999.

[100] L. P. Maneski, N. Jorgovanović, V. Ilić, S. Došen, T. Keller, M. B. Popović, and D. B. Popović, “Electrical stimulation for the suppression of pathological tremor,” *Medical & biological engineering & computing*, vol. 49, no. 10, pp. 1187–1193, 2011.

[101] G. Grimaldi and M. Manto, ““old” and emerging therapies of human tremor,” *Clinical Medicine Insights: Therapeutics*, vol. 2, pp. 1–10, 2010.

[102] G. Deuschl, J. Raethjen, H. Hellriegel, and R. Elble, “Treatment of patients with essential tremor,” *The Lancet Neurology*, vol. 10, no. 2, pp. 148–161, 2011.

[103] W. Wang, L. Xie, and X. Wang, “Tremor detection using smartphone-based acoustic sensing,” in *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*. ACM, 2017, pp. 309–312.

[104] W. Zeng, H. Su, H. Zhong, H. Cui, L. Sun, and L. Xia, “An optimized method for the limb tremor measurement,” in *Proceedings of the 2nd International Conference on Medical and Health Informatics*. ACM, 2018, pp. 65–68.

[105] S. A. Shah, G. Tinkhauser, C. C. Chen, S. Little, and P. Brown, “Parkinsonian tremor detection from subthalamic nucleus local field potentials for closed-loop deep brain stimulation,” in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2018, pp. 2320–2324.

[106] D. Impedovo, G. Pirlo, and G. Vessio, “Dynamic handwriting analysis for supporting earlier parkinson’s disease diagnosis,” *Information*, vol. 9, no. 10, p. 247, 2018.

[107] M. Lang, F. Pfister, J. Frohner, K. Abedinpour, D. Pichler, U. Fietzek, T. Um, D. Kulic, S. Endo, and S. Hirche, “A multi-layer gaussian process for motor symptom estimation in people with parkinson’s disease,” *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 11, pp. 3038–3049, 2019.

[108] L. Parisi, N. RaviChandran, and M. L. Manaog, “Feature-driven machine learning to improve early diagnosis of parkinson’s disease,” *Expert Systems with Applications*, vol. 110, pp. 182–190, 2018.

[109] S. Lahmiri and A. Shmuel, “Detection of parkinson’s disease based on voice patterns ranking and optimized support vector machine,” *Biomedical Signal Processing and Control*, vol. 49, pp. 427–433, 2019.

[110] C. O. Sakar, G. Serbes, A. Gunduz, H. C. Tunc, H. Nizam, B. E. Sakar, M. Tutuncu, T. Aydin, M. E. Isenkul, and H. Apaydin, “A comparative analysis of speech signal processing algorithms for parkinson’s disease classification and the use of the tunable q-factor wavelet transform,” *Applied Soft Computing*, vol. 74, pp. 255–263, 2019.

[111] S. Grover, S. Bhartia, A. Yadav, K. Seeja *et al.*, “Predicting severity of parkinson’s disease using deep learning,” *Procedia Computer Science*, vol. 132, pp. 1788–1794, 2018.

[112] E. Rastegari, S. Azizian, and H. Ali, “Machine learning and similarity network approaches to support automatic classification of parkinson’s diseases using accelerometer-based gait analysis,” in *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019, pp. 4231–4242.

[113] M. D. Hssayeni, J. Jimenez-Shahed, and B. Ghoraani, “Hybrid feature extraction for detection of degree of motor fluctuation severity in parkinson’s disease patients,” *Entropy*, vol. 21, no. 2, p. 137, 2019.

[114] F. Rahimi, O. Samotus, J. Lee, and M. Jog, “Effective management of upper limb parkinsonian tremor by incobotulinumtoxina injections using sensor-based biomechanical patterns,” *Tremor and Other Hyperkinetic Movements*, vol. 5, 2015.

[115] S. Memar, M. Delrobaei, M. Pieterman, K. McIsaac, and M. Jog, “Quantification of whole-body bradykinesia in parkinson’s disease participants using multiple inertial sensors,” *Journal of the neurological sciences*, vol. 387, pp. 157–165, 2018.

[116] M. Delrobaei, S. Tran, G. Gilmore, K. McIsaac, and M. Jog, “Characterization of multi-joint upper limb movements in a single task to assess bradykinesia,” *Journal of the neurological sciences*, vol. 368, pp. 337–342, 2016.

[117] M. Delrobaei, S. Memar, M. Pieterman, T. W. Stratton, K. McIsaac, and M. Jog, “Towards remote monitoring of parkinson’s disease tremor using wearable motion capture systems,” *Journal of the neurological sciences*, vol. 384, pp. 38–45, 2018.

[118] O. Samotus, F. Rahimi, J. Lee, and M. Jog, "Functional ability improved in essential tremor by incobotulinumtoxina injections using kinematically determined biomechanical patterns–a new future," *PloS one*, vol. 11, no. 4, pp. 1–17, 2016.

[119] S. G. Yuen, D. T. Kettler, P. M. Novotny, R. D. Plowes, and R. D. Howe, "Robotic motion compensation for beating heart intracardiac surgery," *The International Journal of Robotics Research*, vol. 28, no. 10, pp. 1355–1372, 2009.

[120] C. N. Riviere, J. Gangloff, and M. De Mathelin, "Robotic compensation of biological motion to enhance surgical accuracy," *Proceedings of the IEEE*, vol. 94, no. 9, pp. 1705–1716, 2006.

[121] J. R. S. de Koste, J. P. Cuijpers, F. G. Geest, F. J. Lagerwaard, B. J. Slotman, and S. Senan, "Verifying 4d gated radiotherapy using time-integrated electronic portal imaging: a phantom and clinical study," *Radiation Oncology*, vol. 2, no. 1, pp. 1–9, 2007.

[122] G. P. Moustris, S. C. Hiridis, K. M. Deliparaschos, and K. M. Konstantinidis, "Evolution of autonomous and semi-autonomous robotic surgical systems: a review of the literature," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 7, no. 4, pp. 375–392, 2011.

[123] M. Tavakoli, R. Patel, and M. Moallem, "Haptic interaction in robot-assisted endoscopic surgery: a sensorized end-effector," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 1, no. 2, pp. 53–63, 2005.

[124] T. Ortmaier, M. Groger, D. H. Boehm, V. Falk, and G. Hirzinger, "Motion estimation in beating heart surgery," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 10, pp. 1729–1740, 2005.

[125] W. T. Ang, C. N. Riviere, and P. K. Khosla, "An active hand-held instrument for enhanced microsurgical accuracy," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2000, pp. 878–886.

[126] K. Veluvolu and W. Ang, "Estimation and filtering of physiological tremor for real-time compensation in surgical robotics applications," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 6, no. 3, pp. 334–342, 2010.

[127] A. Hossen, M. Muthuraman, J. Raethjen, G. Deuschl, and U. Heute, "Discrimination of parkinsonian tremor from essential tremor by implementation of a wavelet-based soft-decision technique on emg and accelerometer signals," *Biomedical Signal Processing and Control*, vol. 5, no. 3, pp. 181–188, 2010.

[128] L. Ai, J. Wang, and R. Yao, "Classification of parkinsonian and essential tremor using empirical mode decomposition and support vector machine," *Digital Signal Processing*, vol. 21, no. 4, pp. 543–550, 2011.

[129] O. Sushkova, A. Gabova, A. Karabanov, I. Kershner, K. Y. Obukhov, and Y. V. Obukhov, "Time–frequency analysis of simultaneous measurements of electroencephalograms, electromyograms, and mechanical tremor under parkinson disease," *Journal of Communications Technology and Electronics*, vol. 60, no. 10, pp. 1109–1116, 2015.

[130] R. J. Elble, H. Hellriegel, J. Raethjen, and G. Deuschl, "Assessment of head tremor with accelerometers versus gyroscopic transducers," *Movement disorders clinical practice*, vol. 4, no. 2, pp. 205–211, 2017.

[131] L. Launer, K. Berger, M. Breteler, J. Dartigues, M. Baldereschi, L. Fratiglioni, A. Lobo, J. Martinez-Lage, C. Trenkwalder, A. Hofman *et al.*, "Prevalence of parkinson's disease in europe: A collaborative study of population-based cohorts. neurologic diseases in the elderly research group." *Neurology*, vol. 54, no. 11 Suppl 5, pp. S21–3, 2000.

[132] E. D. Louis and J. J. Ferreira, "How common is the most common adult movement disorder? update on the worldwide prevalence of essential tremor," *Movement Disorders*, vol. 25, no. 5, pp. 534–541, 2010.

[133] K.-Y. Kwon, H.-S. Ryu, H. M. Lee, M.-J. Kim, H.-W. Shin, H. K. Park, S. You, Y.-H. Sung, S. J. Chung, and S.-B. Koh, "Hand tremor questionnaire: A useful screening tool for differentiating patients with hand tremor between parkinson's disease and essential tremor," *Journal of Clinical Neurology*, vol. 14, no. 3, pp. 381–386, 2018.

[134] N. H. Ghassemi, F. Marxreiter, C. F. Pasluosta, P. Kugler, J. Schlachetzki, A. Schramm, B. M. Eskofier, and J. Klucken, "Combined accelerometer and emg analysis to differentiate essential tremor from parkinson's disease," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2016, pp. 672–675.

[135] W. Chen, F. Hopfner, J. S. Becktepe, and G. Deuschl, "Rest tremor revisited: Parkinson's disease and other disorders," *Translational neurodegeneration*, vol. 6, no. 16, pp. 1–8, 2017.

[136] G. Rizzo, M. Copetti, S. Arcuti, D. Martino, A. Fontana, and G. Logroscino, "Accuracy of clinical diagnosis of parkinson disease: a systematic review and meta-analysis," *Neurology*, vol. 86, no. 6, pp. 566–576, 2016.

[137] E. D. Louis, N. Hernandez, and M. Michalec, "Prevalence and correlates of rest tremor in essential tremor: cross-sectional survey of 831 patients across four distinct cohorts," *European journal of neurology*, vol. 22, no. 6, pp. 927–932, 2015.

[138] M. A. Thenganatt and E. D. Louis, "Distinguishing essential tremor from parkinson's disease: bedside tests and laboratory evaluations," *Expert review of neurotherapeutics*, vol. 12, no. 6, pp. 687–696, 2012.

[139] K.-Y. Kwon, H. M. Lee, S.-M. Lee, S. H. Kang, and S.-B. Koh, "Comparison of motor and non-motor features between essential tremor and tremor dominant parkinson's disease," *Journal of the neurological sciences*, vol. 361, pp. 34–38, 2016.

[140] Z. Farkas, A. Csillik, I. Szirmai, and A. Kamondi, "Asymmetry of tremor intensity and frequency in parkinson's disease and essential tremor," *Parkinsonism & related disorders*, vol. 12, no. 1, pp. 49–55, 2006.

[141] L. di Biase, J.-S. Brittain, S. A. Shah, D. J. Pedrosa, H. Cagnan, A. Mathy, C. C. Chen, J. F. Martin-Rodriguez, P. Mir, L. Timmerman *et al.*, "Tremor stability index: a new tool for differential diagnosis in tremor syndromes," *Brain*, vol. 140, no. 7, pp. 1977–1986, 2017.

[142] A. Schrag and J. M. Schott, "Epidemiological, clinical, and genetic characteristics of early-onset parkinsonism," *The Lancet Neurology*, vol. 5, no. 4, pp. 355–363, 2006.

[143] S. Jain, S. E. Lo, and E. D. Louis, "Common misdiagnosis of a common neurological disorder: how are we misdiagnosing essential tremor?" *Archives of neurology*, vol. 63, no. 8, pp. 1100–1104, 2006.

[144] E. Tolosa, G. Wenning, and W. Poewe, "The diagnosis of parkinson's disease," *The Lancet Neurology*, vol. 5, no. 1, pp. 75–86, 2006.

[145] H. T. Benamer, J. Patterson, D. G. Grosset, J. Booij, K. De Bruin, E. Van Royen, J. Speelman, M. Horstink, H. Sips, R. Dierckx *et al.*, "Accurate differentiation of parkinsonism and essential tremor using visual assessment of [123i]-fp-cit spect imaging: the [123i]-fp-cit study group," *Movement disorders: official journal of the Movement Disorder Society*, vol. 15, no. 3, pp. 503–510, 2000.

[146] P. S. Group *et al.*, "A multicenter assessment of dopamine transporter imaging with dopascan/spect in parkinsonism," *Neurology*, vol. 55, no. 10, pp. 1540–1547, 2000.

[147] F. Ba and W. W. Martin, "Dopamine transporter imaging as a diagnostic tool for parkinsonism and related disorders in clinical practice," *Parkinsonism & related disorders*, vol. 21, no. 2, pp. 87–94, 2015.

[148] S. R. Suwijn, C. J. van Boheemen, R. J. de Haan, G. Tissingh, J. Booij, and R. M. de Bie, "The diagnostic accuracy of dopamine transporter spect imaging to detect nigrostriatal cell loss in patients with parkinson's disease or clinically uncertain parkinsonism: a systematic review," *EJNMMI research*, vol. 5, no. 1, p. 12, 2015.

[149] A. Hossen, M. Muthuraman, Z. Al-Hakim, J. Raethjen, G. Deuschl, and U. Heute, "Discrimination of parkinsonian tremor from essential tremor using statistical signal characterization of the spectrum of accelerometer signal," *Bio-medical materials and engineering*, vol. 23, no. 6, pp. 513–531, 2013.

[150] A. Antonini, P. Berto, S. Lopatriello, F. Tamma, L. Annemans, and M. Chambers, "Cost-effectiveness of 123i-fp-cit spect in the differential diagnosis of essential

tremor and parkinson's disease in italy," *Movement disorders: official journal of the Movement Disorder Society*, vol. 23, no. 15, pp. 2202–2209, 2008.

[151] M. Giuberti, G. Ferrari, L. Contin, V. Cimolin, C. Azzaro, G. Albani, and A. Mauro, "Automatic updrs evaluation in the sit-to-stand task of parkinsonians: Kinematic analysis and comparative outlook on the leg agility task," *IEEE journal of biomedical and health informatics*, vol. 19, no. 3, pp. 803–814, 2015.

[152] F. Parisi, G. Ferrari, M. Giuberti, L. Contin, V. Cimolin, C. Azzaro, G. Albani, and A. Mauro, "Body-sensor-network-based kinematic characterization and comparative outlook of updrs scoring in leg agility, sit-to-stand, and gait tasks in parkinson's disease," *IEEE journal of biomedical and health informatics*, vol. 19, no. 6, pp. 1777–1793, 2015.

[153] H. F. Nweke, Y. W. Teh, M. A. Al-Garadi, and U. R. Alo, "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges," *Expert Systems with Applications*, vol. 105, pp. 233–261, 2018.

[154] N. Kostikis, D. Hristu-Varsakelis, M. Arnaoutoglou, and C. Kotsavasiloglou, "A smartphone-based tool for assessing parkinsonian hand tremor," *IEEE journal of biomedical and health informatics*, vol. 19, no. 6, pp. 1835–1842, 2015.

[155] S. Patel, K. Lorincz, R. Hughes, N. Huggins, J. Growdon, D. Standaert, M. Akay, J. Dy, M. Welsh, and P. Bonato, "Monitoring motor fluctuations in patients with parkinson's disease using wearable sensors," *IEEE transactions on information technology in biomedicine*, vol. 13, no. 6, pp. 864–873, 2009.

[156] S. Reinfelder, R. Hauer, J. Barth, J. Klucken, and B. M. Eskofier, "Timed up-and-go phase segmentation in parkinson's disease patients using unobtrusive inertial sensors," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2015, pp. 5171–5174.

[157] J. Barth, J. Klucken, P. Kugler, T. Kammerer, R. Steidl, J. Winkler, J. Hornegger, and B. Eskofier, "Biometric and mobile gait analysis for early diagnosis and therapy monitoring in parkinson's disease," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2011, pp. 868–871.

[158] S. V. Perumal and R. Sankar, "Gait and tremor assessment for patients with parkinson's disease using wearable sensors," *Ict Express*, vol. 2, no. 4, pp. 168–174, 2016.

[159] B. M. Eskofier, S. I. Lee, J.-F. Daneault, F. N. Golabchi, G. Ferreira-Carvalho, G. Vergara-Diaz, S. Sapienza, G. Costante, J. Klucken, T. Kautz *et al.*, "Recent machine learning advancements in sensor-based mobility analysis: Deep learning for parkinson's disease assessment," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2016, pp. 655–658.

[160] X. Zheng, A. Vieira, S. L. Marcos, Y. Aladro, and J. Ordieres-Mere, "Activity-aware essential tremor evaluation using deep learning method based on acceleration data," *Parkinsonism & related disorders*, vol. 58, pp. 17–22, 2019.

[161] E. Rovini, C. Maremmani, and F. Cavallo, "How wearable sensors can support parkinson's disease diagnosis and treatment: a systematic review," *Frontiers in neuroscience*, vol. 11, p. 555, 2017.

[162] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in neural information processing systems*, 1990, pp. 396–404.

[163] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury *et al.*, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal processing magazine*, vol. 29, 2012.

[164] N. Y. Hammerla, J. Fisher, P. Andras, L. Rochester, R. Walker, and T. Plötz, "Pd disease state assessment in naturalistic environments using deep learning," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[165] H. B. Kim, W. W. Lee, A. Kim, H. J. Lee, H. Y. Park, H. S. Jeon, S. K. Kim, B. Jeon, and K. S. Park, "Wrist sensor-based tremor severity quantification in parkinson's disease using convolutional neural network," *Computers in biology and medicine*, vol. 95, pp. 140–146, 2018.

[166] J. Camps, A. Sama, M. Martin, D. Rodriguez-Martin, C. Perez-Lopez, J. M. M. Arostegui, J. Cabestany, A. Catala, S. Alcaine, B. Mestre *et al.*, "Deep learning for freezing of gait detection in parkinson's disease patients in their homes using a waist-worn inertial measurement unit," *Knowledge-Based Systems*, vol. 139, pp. 119–131, 2018.

[167] P. Povalej Bržan, J. A. Gallego, J. P. Romero, V. Glaser, E. Rocon, J. Benito-León, F. Bermejo-Pareja, I. Posada, and A. Holobar, "New perspectives for computer-aided discrimination of parkinson's disease and essential tremor," *Complexity*, vol. 2017, 2017.

[168] S. Barrantes, A. J. S. Egea, H. A. G. Rojas, M. J. Marti, Y. Compta, F. Valldeoriola, E. S. Mezquita, E. Tolosa, and J. Valls-Sole, "Differential diagnosis between parkinson's disease and essential tremor using the smartphone's accelerometer," *PloS one*, vol. 12, no. 8, p. e0183843, 2017.

[169] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[170] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[171] D. Ravi, C. Wong, F. Deligianni, M. Berthelot, J. Andreu-Perez, B. Lo, and G.-Z. Yang, "Deep learning for health informatics," *IEEE journal of biomedical and health informatics*, vol. 21, no. 1, pp. 4–21, 2016.

[172] I. Goodfellow, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[173] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[174] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, 2014.

[175] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *NIPS 2014 Workshop on Deep Learning*, 2014.

[176] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Advances in neural information processing systems*, 2006, pp. 1473–1480.

[177] E. P. Xing, M. I. Jordan, S. J. Russell, and A. Y. Ng, "Distance metric learning with application to clustering with side-information," in *Advances in neural information processing systems*, 2003, pp. 521–528.

[178] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 209–216.

[179] Y. Liu, H. Li, and X. Wang, "Rethinking feature discrimination and polymerization for large-scale recognition," *arXiv preprint arXiv:1710.00870*, 2017.

[180] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a" siamese" time delay neural network," in *Advances in neural information processing systems*, 1994, pp. 737–744.

[181] A. Medela, A. Picon, C. L. Saratxaga, O. Belar, V. Cabezón, R. Cicchi, R. Bilbao, and B. Glover, "Few shot learning in histopathological images: reducing the need of labeled data on biological datasets," in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. IEEE, 2019, pp. 1860–1864.

[182] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2. Lille, 2015.

[183] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[184] K. Roth, T. Milbich, S. Sinha, P. Gupta, B. Ommer, and J. P. Cohen, "Revisiting training strategies and generalization performance in deep metric learning," *arXiv preprint arXiv:2002.08473*, 2020.

[185] J. Hu, J. Lu, and Y.-P. Tan, "Discriminative deep metric learning for face verification in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1875–1882.

[186] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 212–220.

[187] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[188] A. Sanakoyeu, V. Tschernezki, U. Buchler, and B. Ommer, "Divide and conquer the embedding space for metric learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 471–480.

[189] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh, "No fuss distance metric learning using proxies," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 360–368.

[190] K. Roth, B. Brattoli, and B. Ommer, "Mic: Mining interclass characteristics for improved metric learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 8000–8009.

[191] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krahenbuhl, "Sampling matters in deep embedding learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2840–2848.

[192] X. Lin, Y. Duan, Q. Dong, J. Lu, and J. Zhou, "Deep variational metric learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 689–704.

[193] H. Xuan, R. Souvenir, and R. Pless, "Deep randomized ensembles for metric learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 723–734.

[194] P. Jacob, D. Picard, A. Histace, and E. Klein, "Metric learning with horde: High-order regularizer for deep embeddings," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6539–6548.

[195] K. Roth, T. Milbich, and B. Ommer, "Pads: Policy-adapted sampling for visual similarity learning," *arXiv preprint arXiv:2003.11113*, 2020.

[196] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Advances in neural information processing systems*, 2016, pp. 1857–1865.

[197] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 539–546.

[198] C. Luo, J. Zhan, X. Xue, L. Wang, R. Ren, and Q. Yang, "Cosine normalization: Using cosine similarity instead of dot product in neural networks," in *International Conference on Artificial Neural Networks*. Springer, 2018, pp. 382–391.

[199] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[200] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[201] Y. Zheng, D. K. Pal, and M. Savvides, "Ring loss: Convex feature normalization for face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5089–5097.

[202] A. Singhal *et al.*, "Modern information retrieval: A brief overview," *IEEE Data Eng. Bull.*, vol. 24, no. 4, pp. 35–43, 2001.

[203] R. Ranjan, C. D. Castillo, and R. Chellappa, "L2-constrained softmax loss for discriminative face verification," *arXiv preprint arXiv:1703.09507*, 2017.

[204] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "Normface: L2 hypersphere embedding for face verification," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 1041–1049.

[205] S. Gopal and Y. Yang, "Von mises-fisher clustering models," in *International Conference on Machine Learning*, 2014, pp. 154–162.

[206] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5265–5274.

[207] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[208] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699.

[209] X. Zhang, R. Zhao, Y. Qiao, X. Wang, and H. Li, "Adacos: Adaptively scaling cosine logits for effectively learning deep face representations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 823–10 832.

[210] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 1735–1742.

[211] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, “Large scale online learning of image similarity through ranking,” *Journal of Machine Learning Research*, vol. 11, no. Mar, pp. 1109–1135, 2010.

[212] P. S. Addison, “Wavelet transforms and the ecg: a review,” *Physiological measurement*, vol. 26, no. 5, p. R155, 2005.

[213] I. Daubechies, *Ten lectures on wavelets*. SIAM, 1992.

[214] R. E. Kalman *et al.*, “A new approach to linear filtering and prediction problems,” *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[215] G. Bishop and G. Welch, “An introduction to the kalman filter,” *Proc of SIGGRAPH, Course*, vol. 8, no. 27599-23175, p. 41, 2001.

[216] M. Liu, D. Zhang, S. Chen, and H. Xue, “Joint binary classifier learning for ecoc-based multi-class classification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 11, pp. 2335–2341, 2016.

[217] N. Eghbali and G. A. Montazer, “Improving multiclass classification using neighborhood search in error correcting output codes,” *Pattern Recognition Letters*, vol. 100, pp. 74–82, 2017.

[218] M. Tangermann, K.-R. Müller, A. Aertsen, N. Birbaumer, C. Braun, C. Brunner, R. Leeb, C. Mehring, K. J. Miller, G. Mueller-Putz *et al.*, “Review of the bci competition iv,” *Frontiers in neuroscience*, vol. 6, p. 55, 2012.

[219] T. G. Dietterich and G. Bakiri, “Solving multiclass learning problems via error-correcting output codes,” *Journal of artificial intelligence research*, vol. 2, pp. 263–286, 1994.

[220] A. Mohammadi and A. Asif, “Distributed consensus + innovation particle filtering for bearing/range tracking with communication constraints,” *IEEE Transactions on Signal Processing*, vol. 63, no. 3, pp. 620–635, 2015.

[221] N. Kwak and C.-H. Choi, “Input feature selection by mutual information based on parzen window,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 12, pp. 1667–1671, 2002.

[222] R. T. Schirrmeister, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball, “Deep learning with convolutional neural networks for eeg decoding and visualization,” *Human brain mapping*, vol. 38, no. 11, pp. 5391–5420, 2017.

[223] T. Tanaka and M. Arvaneh, *Signal processing and machine learning for brain-machine interfaces*. Institution of Engineering & Technology, 2018.

[224] A. Patanè and M. Kwiatkowska, "Calibrating the classifier: Siamese neural network architecture for end-to-end arousal recognition from ecg," in *International Conference on Machine Learning, Optimization, and Data Science*. Springer, 2018, pp. 1–13.

[225] Z. Alaverdyan, J. Jung, R. Bouet, and C. Lartizien, "Regularized siamese neural network for unsupervised outlier detection on brain multiparametric magnetic resonance imaging: application to epilepsy lesion screening," *Medical Image Analysis*, vol. 60, p. 101618, 2020.

[226] A. Fares, S.-h. Zhong, and J. Jiang, "Eeg-based image classification via a region-level stacked bi-directional deep learning framework," *BMC Medical Informatics and Decision Making*, vol. 19, no. 6, p. 268, 2019.

[227] M. Venturelli, G. Borghi, R. Vezzani, and R. Cucchiara, "From depth data to head pose estimation: a siamese approach," *arXiv preprint arXiv:1703.03624*, 2017.

[228] S. Stober, "Learning discriminative features from electroencephalography recordings by encoding similarity constraints," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 6175–6179.

[229] E. Maiorana, "Eeg-based biometric verification using siamese cnns," in *International Conference on Image Analysis and Processing*. Springer, 2019, pp. 3–11.

[230] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[231] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Multiclass brain–computer interface classification by riemannian geometry," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 4, pp. 920–928, 2011.

[232] X. Navarro-Sune, A. Hudson, F. D. V. Fallani, J. Martinerie, A. Witon, P. Pouget, M. Raux, T. Similowski, and M. Chavez, "Riemannian geometry applied to detection of respiratory states from eeg signals: The basis for a brain–ventilator interface," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 5, pp. 1138–1148, 2016.

[233] X. Xie, Z. L. Yu, H. Lu, Z. Gu, and Y. Li, "Motor imagery classification based on bilinear sub-manifold learning of symmetric positive-definite matrices," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 6, pp. 504–516, 2016.

[234] N. Robinson, A. P. Vinod, K. K. Ang, K. P. Tee, and C. T. Guan, "Eeg-based classification of fast and slow hand movements using wavelet-csp algorithm," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 8, pp. 2123–2132, 2013.

[235] Y. Dai, X. Zhang, Z. Chen, and X. Xu, "Classification of electroencephalogram signals using wavelet-csp and projection extreme learning machine," *Review of Scientific Instruments*, vol. 89, no. 7, p. 074302, 2018.

[236] R. C. Guido, "Effectively interpreting discrete wavelet transformed signals [lecture notes]," *IEEE Signal Processing Magazine*, vol. 34, no. 3, pp. 89–100, 2017.

[237] O. Rioul and M. Vetterli, "Wavelets and signal processing," *IEEE signal processing magazine*, vol. 8, no. ARTICLE, pp. 14–38, 1991.

[238] K. C. Veluvolu, U.-X. Tan, W. T. Latt, C. Shee, and W. T. Ang, "Bandlimited multiple fourier linear combiner for real-time tremor compensation," in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2007, pp. 2847–2850.

[239] G. N. Reeke, R. R. Poznanski, K. A. Lindsay, J. R. Rosenberg, and O. Sporns, *Modeling in the neurosciences: from biological systems to neuromimetic robotics*. CRC Press, 2005.

[240] H. A. G. Rojas, P. C. Cuevas, E. E. Z. Figueras, S. C. Foix, and A. J. S. Egea, "Time measurement characterization of stand-to-sit and sit-to-stand transitions by using a smartphone," *Medical & biological engineering & computing*, vol. 56, no. 5, pp. 879–888, 2018.

[241] S. F. Atashzar, I. G. Polushin, and R. V. Patel, "A small-gain approach for nonpassive bilateral telerobotic rehabilitation: Stability analysis and controller synthesis," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 49–66, 2016.

[242] M. Shahbazi, S. F. Atashzar, and R. V. Patel, "A systematic review of multilateral teleoperation systems," *IEEE transactions on haptics*, vol. 11, no. 3, pp. 338–356, 2018.

[243] S. F. Atashzar, M. Shahbazi, M. Tavakoli, and R. V. Patel, "A passivity-based approach for stable patient–robot interaction in haptics-enabled rehabilitation systems: modulated time-domain passivity control," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 991–1006, 2016.

[244] ——, "A grasp-based passivity signature for haptics-enabled human-robot interaction: Application to design of a new safety mechanism for robotic rehabilitation," *The International Journal of Robotics Research*, vol. 36, no. 5-7, pp. 778–799, 2017.

[245] ——, "A new passivity-based control technique for safe patient-robot interaction in haptics-enabled rehabilitation systems," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 4556–4561.

[246] A. V. Oppenheim and R. W. Schafer, *Discrete-time signal processing*. Pearson Education, 2014.

[247] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[248] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations (ICLR)*, 2015.

[249] E. S. Pearson, R. B. D "'AGOSTINO, and K. O. Bowman, "Tests for departure from normality: Comparison of powers," *Biometrika*, vol. 64, no. 2, pp. 231–246, 1977.

[250] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 618–626.

[251] R. Kotikalapudi and contributors, "keras-vis," https://github.com/raghakot/keras-vis, 2017.

[252] E. Smirnov, A. Melnikov, A. Oleinik, E. Ivanova, I. Kalinovskiy, and E. Luckyanets, "Hard example mining with auxiliary embeddings," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 37–46.

[253] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in neural information processing systems*, 2017, pp. 4077–4087.