Self-regulating interval type-2 neuro-fuzzy inference system for non-stationary EEG signal processing

Ankit Kumar Das

2017

Ankit Kumar Das. (2017). Self-regulating interval type-2 neuro-fuzzy inference system for non-stationary EEG signal processing. Doctoral thesis, Nanyang Technological University, Singapore.

<http://hdl.handle.net/10356/69545>

<https://doi.org/10.32657/10356/69545>

Self-regulating Interval Type-2 Neuro-fuzzy Inference System for Non-stationary EEG Signal Processing

Ankit Kumar Das

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

2017

My prayers to God almighty.
Dedicated to my Grandparents and beloved wife.

“The ultimate arbiter of truth is experiment, not the comfort one derives from one’s a priori beliefs, nor the beauty or elegance one ascribes to one’s theoretical models.”

Lawrence M. Krauss

Acknowledgement

It is a great pleasure for me to thank the people who have helped me during my PhD. First and foremost, I express my sincere gratitude and indebtedness to my supervisor Dr. Suresh Sundaram for his guidance, motivation and immense knowledge. Without his consistent support and help, the research would not have been so enriching and fulfilling. I have learned from him the ways of approaching an open problem and analyze it fundamentally. I would also like to express my heartfelt gratitude to Dr. N. Sundararajan who took his valuable time to guide and help me throughout the whole period.

Last but not the least; I would like to thank my grand parents, my parents, my wife Pritee Agrawal for all the help, motivation, love and support that they have provided. It would have been impossible to complete this thesis without the confidence that I had their support, and help, during the entire period of my PhD. Finally, I would like to thank my friends, and colleagues, especially Atul Bhaskar Mishra, Kartick Subramanian, Amiya Patnaik, Arun Kumar and Venugopalan TK, for adding life to the graduation years. I thoroughly enjoyed the time I spent with you all, having various technical and non-technical discussions.

Finally, I would like to thank Center for Computational Intelligence, School of Computer Science and Engineering and Nanyang Technological University for providing me with such an infrastructure and support to conduct my research.

Contents

Acknowledgements	i
Abstract	vii
List of Figures	ix
List of Tables	xi
List of Symbols	xv
List of Abbreviations	xvii
1 Introduction	1
1.1 Background and Motivation	1
1.2 Issues and Challenges	6
1.3 Objectives of the Study	8
1.4 Major Contributions of the Thesis	9
1.5 Thesis Organization	12
2 A Review of Brain Signal Processing Algorithms for Motor-imagery based Brain Computer Interfaces	15
2.1 Frequency Domain Based Approach	16
2.2 Time Domain Based Approach	16
2.3 Time-frequency Based Approach	17
2.4 Spatial Filtering Approach	17
2.4.1 Common Spatial Pattern Algorithm	18
2.4.2 Adaptive CSP Algorithms	20
2.4.3 Regularized CSP Algorithms	21
2.4.4 Summary of Spatial Filtering Approaches	25
2.5 Spectral Filtering Approaches	26

2.5.1	Summary of Spectral Filtering Approaches	33
2.6	Summary	34
3	A Robust Common Spatial Pattern Feature Extraction Algorithm	35
3.1	Illustrative Example for Trial/Channel Selection	37
3.2	Robust Common Spatial Pattern Algorithm	38
3.2.1	Feature Extraction	43
3.3	Data Set Description	43
3.4	Performance Evaluation	44
3.4.1	Performance of RoCSP - BCI Competition IV Data Set-IIa	45
3.4.2	Performance of RoCSP - BCI Competition III Data Set-IVa . . .	53
3.4.3	Illustration of Effect of Non-stationarity in EEG Data on RoCSP	54
3.5	Summary	58
4	A Self-Regulated Interval Type-2 Neuro-Fuzzy Inference System	59
4.1	A Review of Interval Type-2 Neuro-Fuzzy Inference System literature . .	62
4.1.1	Type-1 Fuzzy Sets	62
4.1.2	Type-2 Fuzzy Sets	63
4.1.3	Interval Type-2 Fuzzy Sets	64
4.1.4	Classification Based on the Types of Fuzzy Inference Mechanism .	65
4.1.5	Classification Based on the Types of Learning Modes	68
4.1.6	Classification Based on the Type of Uncertainty being Modeled .	74
4.1.7	Classification Based on the Types of Consequent	75
4.1.8	Classification Based on the Type-Reduction Mechanism	76
4.1.9	Summary of Interval Type-2 Neuro-fuzzy Inference Systems Dis- cussed in Literature	78
4.2	Self-Regulation in Learning	80
4.3	Self-Regulated Interval Type-2 Neuro-Fuzzy Inference System	81
4.3.1	Self-Regulated Interval Type-2 Neuro-Fuzzy Inference System Ar- chitecture	82
4.3.2	Self-Regulated Interval Type-2 Neuro-Fuzzy Inference System Learn- ing Algorithm	84

4.4	Performance Evaluation	92
4.4.1	Performance Evaluation of SRIT2NFIS using Benchmark Data Sets	93
4.4.2	Comparison of SRIT2NFIS with Interval Type-2 Algorithms . . .	96
4.5	Summary	96
5	An Intelligent-Brain Computer Interface for Handling Non-stationarities in EEG Signals	99
5.1	Performance Evaluation of I-BCI	100
5.1.1	Performance of I-BCI in Handling Intersession Non-stationarity - BCI Competition IV Data Set-IIa	101
5.1.2	Performance of I-BCI in Handling Intrasection Non-stationarity - BCI Competition III Data Set-IIIa	104
5.1.3	Discussion on the Synergy of the Hybrid and the Contribution of Individual Components, RoCSP and SRIT2NFIS	105
5.2	Summary	108
6	A Discriminative Subject-Specific Spatio-Spectral Filter Selection Approach for EEG based Motor-Imagery Task Classification	109
6.1	Spatio-Spectral Filter Selection using SRIT2NFIS	113
6.1.1	Feature Generation from EEG using a Filter Bank	114
6.1.2	SRIT2NFIS Motor-imagery Task Classifier	115
6.1.3	Spectral Filters Selection using Self-Regulated Interval Type-2 Neuro-Fuzzy Inference System	118
6.1.4	Spatial Filters Selection using Self-Regulated Interval Type-2 Neuro-Fuzzy Inference System	119
6.2	Experimental Results	121
6.2.1	Performance Evaluation of S-IBCI and SS-IBCI using the BCI Competition IV Data Set-IIa	122
6.2.2	Performance Evaluation of S-IBCI and SS-IBCI using the BCI Competition III Data Set-IVa	125
6.3	Summary	127

7	Realization of I-BCI for Control of a Quadcopter	129
7.1	A Review of BCI Systems for Device Control	129
7.2	Realization of I-BCI for Quadcopter Control	130
7.2.1	Data Collection and Pre-processing	131
7.2.2	Quadcopter Model	133
7.2.3	Experimental Setup	134
7.3	Performance Evaluation	136
7.3.1	Performance Evaluation - Left-Right Motor-imagery Task Classifi- cation	136
7.3.2	Experimental Evaluation - Movement Control of Quadcopter . . .	137
7.4	Summary	140
8	Conclusions and Future Directions	141
8.1	Thesis Summary	141
8.1.1	Robust Common Spatial Pattern Feature Extraction Algorithm .	142
8.1.2	Self-Regulated Interval Type-2 Neuro-Fuzzy Inference System . .	143
8.1.3	Intelligent-Brain Computer Interface for Handling Non-stationarities in EEG Signals	143
8.1.4	Discriminative Subject-Specific Spatio-Spectral Filter Selection Ap- proach for EEG based Motor-Imagery Task Classification	144
8.1.5	Realization of I-BCI for Control of Quadcopter	144
8.2	Future Directions	145
8.2.1	Learning Algorithm	145
8.2.2	Fuzzy Inference Mechanism	145
8.2.3	BCI Systems	146
	Appendix1	148
	List of Publications	151
	References	153

Abstract

Motor-imagery based Brain Computer Interface (BCI) provides a direct communication pathway between the brain and a computer based on the neural activities generated by the brain. Such a technology enables people with physical disabilities to communicate with the external world without using their peripheral nerves and muscles. Moreover, BCI systems can also be used in field of gaming, robotics and human-computer interaction, in general. A typical motor-imagery based BCI system consists of a brain activity acquisition phase to obtain ElectroEncephaloGram (EEG) data using multiple channels, a brain signal processing phase to decode the user intentions and a translation phase for transforming the decoded information to control external devices. However, the usage of multiple channels may lead to several issues: a) Longer preparation time; b) Redundant channels with noisy information. In addition, the thoughts of users vary, which may change the data distribution over time (leads to non-stationary nature of EEG). Some of the trials are also affected by the presence of artifacts in EEG data. Further, the frequency response for a motor-imagery task varies for different individuals. These issues can render the BCI system inaccurate with deteriorated performance. The main goal of this thesis is to develop a unified BCI system that address the above mentioned issues.

There are four major contributions in this thesis. The first contribution is development of a Robust Common Spatial Pattern (RoCSP) feature extraction algorithm that eliminates the trials affected by artifacts and discards the redundant channels to improve the classification performance of the BCI system. Experimental results show that RoCSP is able to reduce the number of channels and produce a good classification performance.

Although, RoCSP provides robust features and improves performance, it does not handle non-stationarity in the EEG data. Recently, it has been shown that interval type-2 fuzzy systems are capable of handling non-stationarity. However, there is a challenge

in developing an evolving sequential learning algorithm for interval type-2 fuzzy systems which can learn and evolve the architecture automatically and select the appropriate samples. The second contribution is the development of a Self-Regulated Interval Type-2 Neuro-Fuzzy Inference System (SRIT2NFIS) to handle the non-stationarity in data. A five-layered modified Takagi-Sugeno-Kang interval type-2 fuzzy inference mechanism forms the structure and the learning algorithm is a self-regulatory learning mechanism. The learning starts with zero rules and as a sample is presented to the network, the learning algorithm monitors the knowledge in the sample and decides whether to delete it, learn it or keep it for future use. When a new rule is added to the network, a regularized projection based learning is employed to compute the optimal weights of the network.

The RoCSP feature extraction algorithm is combined with SRIT2NFIS to form an Intelligent-BCI (I-BCI) system. The features generated by the RoCSP algorithm are used as input to the SRIT2NFIS classifier. I-BCI system handles non-stationarity in the RoCSP features by modeling it as uncertainty in the classifier. Experimental results show that I-BCI system is able to handle both intra/intersession non-stationarity.

The I-BCI system employs a broad frequency range for all the subjects. However, it has been shown in the literature that there is a variation in frequency responses of different subjects. Hence, we improve the I-BCI system for individual subjects. The third contribution is the development of a mechanism for selecting spectral (Spectral-I-BCI (S-IBCI)) and spatio-spectral filters (Spatio-Spectral-I-BCI (SS-IBCI)). S-IBCI system selects spectral filters by eliminating redundant bands from a filter bank in an iterative manner. Further, the SS-IBCI system selects spatial filters from the selected spectral filters. Based on the subject-wise studies, the experimental results indicate a significant performance improvement by both the spectral and spatio-spectral filter selection scheme.

Finally, the I-BCI system is realized for control of a quadcopter using brain signals. The quadcopter is equipped with proportional integral derivative controller for motion in x-y plane. Experiments show that the user is able to control the quadcopter successfully. The proposed system improves the accuracy of EEG based BCI highlighting the significance of robust features, handling non-stationarity and providing subject-specific spatio-spectral filters.

List of Figures

1.1	Typical architecture of a motor-imagery based BCI system.	2
1.2	Illustration of trials recorded using multiple channels over intra/intersessions in EEG based BCI system. The noisy trial (Trial 3) is highlighted by a red box.	4
1.3	Illustration of issues and challenges in EEG based BCI system.	5
2.1	Categorization of brain signal processing algorithms.	16
2.2	Schematic diagram of SBCSP.	30
2.3	Schematic diagram of FBCSP.	30
2.4	Schematic diagram of DFBCSP.	31
3.1	Typical architecture of a motor-imagery based BCI system.	36
3.2	Brain signal processing framework.	36
3.3	Illustrative example for trial/channel selection. The blocks show the dot product of trials for each channel. The red ellipses represent the eliminated noisy trials. The blue ellipses represent the eliminated channels.	39
3.4	Scatter plot for train and test data using three most dominant pair of features obtained by RoCSP for subject A8 of left-right classification, BCI competition IV data set-IIa. The plus sign and circle denote the features of the left hand and right hand movement, respectively. The dashed line represents the linear hyperplane obtained by the train data.	56
3.5	Scatter plot for train and test data using three most dominant pair of features obtained by RoCSP for subject A2 of left-right classification, BCI competition IV data set-IIa. The plus sign and circle denote the features of the left hand and right hand movement, respectively. The dashed line represents the linear hyperplane obtained by the train data.	57

4.1	Brain signal processing framework.	59
4.2	A Gaussian type-1 fuzzy membership function.	63
4.3	Foot print of interval type-2 membership function modeled using uncertain mean	65
4.4	Footprint of interval type-2 membership function modeled using uncertain mean.	75
4.5	Interval type-2 fuzzy set with uncertain standard deviation and its foot print of uncertainty.	76
4.6	Nelson and Narens model [1] of meta-cognition.	81
4.7	Structure of self-regulated interval type-2 neuro-fuzzy inference system. .	82
4.8	Flowchart for SRIT2NFIS.	93
5.1	Brain signal processing framework.	99
6.1	Block diagram of two-step spatio-spectral filtering using SRIT2NFIS classifier.	112
6.2	Architecture and learning scheme of SRIT2NFIS.	116
6.3	Schematic diagram of spatio-spectral filtering using SRIT2NFIS classifier. .	120
7.1	Mechanism for quadcopter flight control using the Intelligent-BCI system. .	132
7.2	EMOTIV headset.	133
7.3	Positions of the shifted sensors on the subject	134
7.4	AR.drone 2.0.	135
7.5	Block diagram of quadcopter control system.	135
7.6	Schematic diagram of the experiment. The user moves the quadcopter from point A to B and vice versa.	138
7.7	Quadcopter at the start position.	138
7.8	The figure shows the left movement of quadcopter at four time instances while moving from point A to point B.	139
7.9	The figure shows the right movement of quadcopter at four time instances while moving back from point B to point A.	140
8.1	Influence of tunable parameters on the IRIS classification accuracy. . . .	150
8.2	Rule growth history for IRIS dataset.	150

List of Tables

2.1	Summary of major spatial filtering algorithms discussed in literature review.	27
2.2	Summary of major spectral filtering algorithms discussed in literature review.	33
3.1	Classification performance (Subject Mean (SM) in % and Standard Deviation (SD)) obtained for six classification tasks by using CSP-LDA and RoCSP-LDA algorithm for the BCI competition IV data set-IIa.	46
3.2	Classification performance (SM in % and SD) obtained for each subject for the standard CSP and its variants and RoCSP-LDA on the intersession(different days) for left vs right hand task -BCI competition IV data set-IIa.	47
3.3	Classification performance (SM in % and SD) obtained for each subject for the standard CSP and its variants and RoCSP-LDA on the intersession(different days) for left vs foot task- BCI competition IV data set-IIa.	48
3.4	Classification performance (SM in % and SD) obtained for each subject for the standard CSP and its variants and RoCSP-LDA on the intersession(different days) for left vs tongue task- BCI competition IV data set-IIa.	49
3.5	Classification performance (SM in % and SD) obtained for each subject for the standard CSP and its variants and RoCSP-LDA on the intersession(different days) for right vs foot task- BCI competition IV data set-IIa.	50
3.6	Classification performance (SM in % and SD) obtained for each subject for the standard CSP and its variants and RoCSP-LDA on the intersession(different days) for right vs tongue task- BCI competition IV data set-IIa.	50

3.7	Classification performance (SM in % and SD) obtained for each subject for the standard CSP and its variants and RoCSP-LDA on the intersession(different days) for foot vs tongue task- BCI competition IV data set-IIa.	51
3.8	Comparison of classification performance (SM in % and SD) obtained for six classification tasks for CSP algorithm and its variants with the proposed RoCSP-LDA on the intersession (different days) data set II-a from BCI competition-IV	52
3.9	Classification performance (Subject Mean (SM) in % and Standard Deviation(SD)) with selected channels using RoCSP, classification performance with all channels using CSP, number of selected channels for each subject, threshold δ for the BCI competition III data set-IVa.	53
3.10	Classification performance (SM in % and SD) obtained for each subject for the standard CSP and its variants and RoCSP-LDA on BCI competition III data set-IVa.	54
4.1	Summary of major interval type-2 algorithms discussed in literature review.	79
4.2	Classification data set	94
4.3	Performance evaluation of SRIT2NFIS using benchmark data set	95
4.4	Performance comparison of SRIT2NFIS with existing type-2 systems. . .	97
5.1	Classification performance (Subject Mean(SM) in % and Standard Deviation(SD)) obtained for six classification tasks by using CSP-LDA and I-BCI algorithm for the BCI competition IV data set-IIa.	101
5.2	Comparison of classification performance (SM in % and SD) obtained for six classification tasks for CSP algorithm and its variants with the proposed I-BCI on the intersession (different days) data set II-a from BCI competition-IV	102
5.3	Classification performance (SM in % and SD) obtained for each subject for the TRCSP-SVM and I-BCI on the left vs right hand task -BCI competition IV dataset-IIa.	103

5.4	Mean Rank of the compared algorithms (I-BCI, CSP, CSSP1, CSSP2, GLRCSP, DLCSP: (DLCSPauto, DLCSPdiff, DLCSPcv), SSRCSPP, TRCSP, WTRCSP and SRCSP) across all the fifty four data sets (nine subjects performing six classification tasks).	105
5.5	Detailed results for the Bonferroni Dunn rank test.	106
5.6	Classification performance (SM in % and SD) obtained for each subject for the standard CSP and its variants on the intrasession BCI competition III data set-IIIa.	107
5.7	Classification performance (SM in % and SD) obtained for each subject for the CSP-LDA, TRCSP-LDA, RoCSP-LDA, CSP-SRIT2NFIS and I-BCI on the left vs right hand task -BCI competition IV dataset-IIa.	107
6.1	Classification performance (Subject Mean (SM) in % and Standard Deviation(SD)) of S-IBCI, SS-IBCI and bands obtained for each subject for all the six classification tasks for the BCI competition IV data set-IIa. . . .	124
6.2	Classification performance in %, selected bands and spatial filters obtained for subject A5 for the left-right classification task for the BCI competition IV data set-IIa.	125
6.3	Performance comparison of S-IBCI and SS-IBCI for the BCI competition IV data set-IIa.	125
6.4	Classification performance of S-IBCI, SS-IBCI and Bands obtained for each subject for the BCI competition III data set-IVa.	126
6.5	Performance comparison of S-IBCI and SS-IBCI on BCI competition III data set-IVa.	126
7.1	Performance evaluation (Mean and Standard Deviation (SD)), generated left/right patterns, accuracy for ten runs while performing the experiment.	137

List of Symbols

V	.....	CSP projection matrix
Z	.....	Transformed EEG signals
$\mathbf{x}$	.....	Input feature vector
κ	.....	Overlap factor
ϕ	.....	Membership of a fuzzy rule
ψ	.....	Spherical potential
$\mathbb{R}$	.....	Set of real numbers
E_c^j	.....	EEG data of j -th trial and c -th class
d	.....	effect-size
σ_p	.....	Pooled standard deviation
K	.....	Total number of rules
m	.....	Number of input features
α	.....	Weighted measure of uncertainty
η	.....	Inter-class overlap factor
E	.....	Prediction error
E_d	.....	Sample delete threshold
E_a	.....	Self-adaptive sample addition threshold
E_l	.....	Self-adaptive sample update threshold
μ	.....	Center of a fuzzy rule
σ	.....	Width of a fuzzy rule
γ	.....	Slope parameter
W	.....	Weight matrix
β	.....	Contribution of a rule to the total output
E_p	.....	Rule pruning threshold

E_S	Novelty threshold
B_i	Band i
f	Spatial filter set

List of Abbreviations

AWSSP	AdaptivelyWeighted Spectral-Spatial Pattern
BCI	Brain Computer Interface
BC	Breast Cancer
BSSFO	Bayesian Spatio-Spectral Filter Optimization
CCSP1	Composite CSP1
CCSP2	Composite CSP2
CD	Critical Difference
CSP	Common Spatial Patterns
CSSP	Common Spatio-Spectral Pattern
CSSSP	Common Sparse Spectral Spatial Pattern
DFBCSP	Discriminative Filter Bank CSP
DLCSP	Regularized CSP with Diagonal Loading
eClass	Evolving Fuzzy Rule based Classifier
EEG	ElectroEncephaloGram
ERD	Event-Related De-synchronization
ERS	Event-Related Synchronization
eT2FIS	evolving Interval Type-2 Fuzzy Inference System
FBCSP	Filter Bank CSP
fMRI	functional Magnetic Resonance Imaging
FOU	Footprint Of Uncertainty
GLRCSP	Regularized CSP with Generic Learning
I-BCI	Intelligent-BCI
ICSP	Invariant CSP
ION	Ionosphere

IS	Image Segmentation
ISSPL	Iterative Spatio-Spectral Patterns Learning algorithm
IT2FNN	Interval Type-2 Fuzzy Neural Network
IT2FNN-SVR ..	Interval Type-2 Fuzzy-Neural Network with Support Vector Regression
IT2NFC-OL ...	Interval Type-2 Neural Fuzzy Chip with On-chip Incremental Learning
IT2NFC-SMM ...	Interval Type-2 Neural Fuzzy Classifier learned through Soft Margin Minimization
IT2NFIS	Interval Type-2 Neuro-Fuzzy Inference System
KL	Kullback-Leibler
KLCSP	Kullback-Leibler CSP
K-M	Karnik-Mendel
LDA	Linear Discriminant Analysis
McFIS	Meta-cognitive Neuro-fuzzy Inference System
McIT2FIS	Meta-cognitive Interval Type-2 Neuro-Fuzzy Inference System
MEG	MagnetoEncephaloGraphy
MMISS	Maximum Mutual Information Spatial Spectral Features
MReIT2NFS	Mutually Recurrent Interval Type-2 Neural Fuzzy System
MRIT2NFS-WB ..	Mamdani-Type Reduced Interval Type-2 Neural Fuzzy System using Weighted Bound-set boundaries
ORGQACO	Online Rule Generation and Q-value-aided Ant Colony Optimization
OS-F-ELM	On-line Sequential Fuzzy Extreme Learning Machine
OSSFN	Optimal Spatio-Spectral Filter Network
OSSFNwDFBCSP	OSSFN utilizing DFBCSP
OSSFNwFBCSP	OSSFN utilizing FBCSP
PET	Positron Emission Tomography
PID	Proportional Integral Derivative
PIMA	PIMA Indian diabetes
RIT2NFS-WB	Reduced Interval Type-2 Neural Fuzzy System using Weighted Bound-set boundaries
RoCSP	Robust Common Spatial Pattern
RPBL	Regularized Projection Based Learning

RSEIT2FNN	Recurrent Self-Evolving Interval Type-2 Fuzzy Neural Network
SAFIS	Sequential Adaptive Fuzzy Inference System
SBCSP	Sub-Band Common Spatial Pattern
sCSP	stationary CSP
SD	Standard Deviation
SEIT2FNN	Self Evolving Interval Type-2 Fuzzy Neural Network
S-IBCI	Spectral-I-BCI
SIT2FNN	Simplified Interval Type-2 Fuzzy Neural Network
SM	Subject Mean
SPEC-CSP	Spectrally-weighted CSP
SRCSP	Spatially Regularized CSP
SRIT2NFIS	Self-Regulated Interval Type-2 Neuro-Fuzzy Inference System
SS-IBCI	Spatio-Spectral-I-BCI
SSRCSP	Regularized CSP with Selected Subjects
SVM	Support Vector Machine
T2SONFS	Type-2 Self-Organizing Neural Fuzzy System
TRCSP	CSP with Tikhonov Regularization
TSCIT2FNN	TSK-Type-Based Self-Evolving Compensatory Interval Type-2 Fuzzy Neural Network
TSK	Takagi-Sugeno-Kang
WTRCSP	CSP with Weighted Tikhonov Regularization

Chapter 1

Introduction

1.1 Background and Motivation

People affected by diseases such as multiple sclerosis, muscular dystrophies, brainstem stroke, Parkinson's disease, etc. often lose control over their muscles and struggle to perform motor activities. Severely affected people may lose their muscle control completely. Some assistive communication technologies such as eye trackers, walkers, etc. have been proposed in the literature [2–4], but they are unable to restore motor function and quality of life in most patients. Hence, there is a need to develop effective ways to improve quality of life for people with motor disabilities.

Brain Computer Interface (BCI) is one of the most popular and effective technologies that provides a means of communication for patients who do not have enough control over their motor system (muscles) to communicate in a normal way. It provides a direct communication path between the brain and the external devices based on various neural activities generated by the brain [5]. It does not rely, in any way, on muscular control. Signals generated by imagination of motor activities are translated into user commands, enabling people with disabilities to communicate with a computer using their thoughts [6, 7]. These systems are called motor-imagery based BCI systems.

The architecture of a typical motor-imagery based BCI system is shown in Fig. 1.1. As seen from the Fig. 1.1, instead of depending on peripheral nerves and muscles, a BCI system directly measures brain activity associated with the user's intent and translates the recorded brain activity into corresponding control signals for BCI applications. The core components of a typical BCI system are explained as follows:

Figure 1.1: Typical architecture of a motor-imagery based BCI system.

- *Brain activity acquisition:* In this block, the brain activity is recorded using different types of sensors. The recorded brain signals are amplified, digitized and then used as input to the BCI system.
- *Brain signal processing:*
  - *Pre-processing block:* This block processes the brain signals in order to filter the relevant information hidden in the EEG signals.
  - *Feature extraction block:* This block transforms the EEG signals to relevant features that describe the underlying neurological process and can be used for classification.
  - *Classification block:* This block is responsible for identifying the user's intent by using the features generated by the feature extraction block.
- *Translation block:* This block is responsible for transforming the output of the classifier into control signals for different external devices.
- *Output device:* The output device can be a computer, a wheelchair, a robotic arm, etc. The output of the classifier is used as a command to control the output device.

- *Feedback*: The feedback can be in visual, auditory or tactile form, and helps the user control his brain activities and adapt according to the feedback in order to improve the performance of BCI system.

The major components of a BCI system are brain activity acquisition and signal processing. First, brain activity acquisition is explained, followed by the signal processing component. Brain activity is generated by the firing of millions of neurons inside the cerebral cortex. Different techniques used to measure the brain activity can be categorized into two groups: invasive and non-invasive [8]. Invasive techniques record data directly from the skull and require surgical interventions to implant electrodes. Implanted electrodes give better signal quality and spatial resolution than non-invasive methods. Early studies in invasive BCI were carried out using primates [9]. Recent studies have shown the feasibility of invasive BCI systems in humans [10, 11]. But, the main drawback of the invasive BCI is that the implanted electrodes have a limited lifetime, which necessitates regular surgery operations for electrode replacement that might be harmful to the health of the subject. Therefore, non-invasive methods are preferred in BCI systems rather than invasive measurement techniques [12]. These non-invasive methods acquire data through electrodes placed on the surface of the head. A number of non-invasive methods explored in BCI applications include functional Magnetic Resonance Imaging (fMRI), Positron Emission Tomography (PET), MagnetoEncephaloGraphy (MEG) and ElectroEncephaloGram (EEG). In most of the current motor-imagery based BCI systems, brain signals are measured using EEG signals due to their high temporal resolution and ease of use [5]. Furthermore, EEG is also a relatively low cost method when compared to methods like PET [13], MEG [14] or fMRI [15] which require expensive equipments and skilled professionals to operate. A montage of electrodes (channels) is placed on the scalp to record the EEG activity. EEG data is recorded from multiple channels on trial by trial basis. The subject is presented with a visual cue and asked to perform imagination of motor tasks. Fig. 1.2¹ shows the EEG data from M channels recorded over n trials of two different classes. These trials are conducted in the same session (e.g., Trials 1 and 2) as well as across sessions (e.g., Trials 1 and 3). The sessions can be conducted on same day as well as on different days.

---

¹Fig. 1.2 is only for illustration and not obtained using experiments.

Figure 1.2: Illustration of trials recorded using multiple channels over intra/interessions in EEG based BCI system. The noisy trial (Trial 3) is highlighted by a red box.

In the signal processing phase, the EEG signals are used for feature extraction and classification of trials. First, the data is spectrally filtered using a frequency range of 8-30 Hz , since this range covers both, the mu and beta ranges that include the frequencies for performing motor imagery tasks [16]. The next step is to extract features that maximize the separation of the two classes of data. Spatial filtering methods like Common Spatial Patterns (CSP) has been used in the literature for feature extraction [17]. In the final step of the signal processing phase, a classifier that differentiates the two classes is learnt.

However, EEG signals suffer from the presence of artifacts in the data as illustrated in Fig. 1.3a². The EEG signals are recorded trial by trial at different timestamps. It can be seen from the Fig. 1.3a that the trial at time instance $2T$ (highlighted by a red box)

²Fig. 1.3 is only for illustration and not obtained using experiments.

(a) Noisy channel and trial. The noisy trial is highlighted by a red box.

(b) Intra/Intersession non-stationarity.

(c) Subject-specific frequency ranges.

Figure 1.3: Illustration of issues and challenges in EEG based BCI system.

is affected by artifacts. The artifacts such as swallowing, blinking or yawning with more signal power than that of the BCI related activity affect some trials, hence, considering all the trials for feature extraction might lead to features that are not robust. Further, it can also be seen from Fig. 1.3a that the channel Ch_2 is noisy. Therefore, considering data from large number of channels may include noisy and redundant channels.

Another issue is the presence of inherent non-stationarity in EEG signals. The trials are conducted session by session, that are carried out on the same day as well as on different days. Non-stationarity in EEG data arises from the fact that there are variations in EEG signals for the same task in two different trials conducted within a session and also across sessions. This is illustrated in the Fig. 1.3b. Variations arise due to delays in the response to visual cues, change of visual tasks, closing of eyes, change of location while placing the electrodes, etc. These variations in the signal can deteriorate the performance of models that assume the stationarity of data.

Further, the frequency ranges used for spectral filtering of the EEG data are subject-specific. It can be seen from Fig. 1.3c that different subjects respond in different ranges of frequencies. Hence, it is necessary to determine a separate set of spatio-spectral filters for an individual subject.

In order to develop an effective EEG based BCI system, the above mentioned issues have to be addressed. This thesis shall deal with the highlighted issues and propose an effective EEG based BCI system. Next, the issues and challenges in this thesis are outlined.

1.2 Issues and Challenges

This section discusses the major issues and challenges in the development of an effective BCI system.

- **Robust Feature Extraction:**

The purpose of BCI systems is to translate the subject's intentions into user commands. Reliable feature extraction from the EEG data is essential for an efficient BCI system. Common Spatial Pattern (CSP) is an effective feature extraction approach, commonly used in this area that computes spatial filters from the raw EEG

data, such that the separation between two classes is maximized [17, 18]. However, the presence of artifacts in the EEG data and noisy channels (Fig. 1.3a) greatly influence the effectiveness of CSP, since, it considers all the trials and all the channels for feature extraction. Considering all the trials might lead to a poor estimation of the class covariance matrices, as some of the trials may be affected by artifacts and have more signal power than BCI related activity [19–22]. Further, employing a large number of channels may cause redundancy of data, inclusion of noisy channels and longer preparation time (which may be inconvenient for the subject using the BCI system). Hence, there is a need to develop a robust feature extraction algorithm that eliminates trials affected by artifacts and selects the appropriate number of channels that provide good classification performance.

- **Handling Non-stationarity:**

The presence of inherent non-stationarity in EEG trials recorded within a session and across sessions deteriorate the effectiveness of a BCI system. Handling non-stationarity is of utmost importance for an accurate BCI system. Non-stationarity in the EEG signals arises due to the trial-to-trial variation for a task within a session and also variations that arise across different sessions. This is illustrated in Fig. 1.3b. These variations are due to the delays in responses to the visual cues, change of visual tasks, closing of eyes, change in placement of electrodes, etc. [23]. These variations deteriorate the performance of most of the models that inherently assume that data is stationary [24–26]. Most algorithms in the literature either adapt the classifier/features by using data from the test set or regularize the CSP algorithm to handle non-stationarity. Even though these algorithms handle non-stationarity to a certain extent, the presence of subject-specific variability in the EEG data influences their performance significantly. Hence, there is need to develop an effective mechanism to handle non-stationarity in EEG data.

- **Online/Sequential Adaptive Framework:**

In EEG based BCI systems, the EEG trials arrive sequentially. Moreover, the data is temporal in nature, which needs to be captured efficiently. Hence, there is a need to develop an algorithm that learns the trials (samples) one-by-one and evolves

over time. In addition, since the data arrives one-at-a-time, the algorithm must be able to select the appropriate samples for learning such that over-training and outliers in the training data can be avoided. Further, non-stationarity in EEG data could change gradually or suddenly which requires the algorithm to be evolving and adaptive in nature.

- **Identification of Subject-Specific Spatio-Spectral Filters:**

Motor-imagery tasks generate Event-Related Synchronization (ERS) and Event-Related De-synchronization (ERD) in certain subject-specific frequency ranges of EEG signals. These frequency ranges vary for different subjects as illustrated in Fig. 1.3c. The selection of frequency ranges for each subject is important for obtaining better classification accuracy of motor-imagery based BCI. Further, the spatial filters extracted corresponding to the selected spectral ranges also influence the classification accuracy. Hence, there is a need to develop a mechanism that finds subject-specific spatio-spectral filters for accurate motor-imagery task classification.

1.3 Objectives of the Study

This thesis addresses the above mentioned issues and challenges in developing a BCI system that has the ability to classify the EEG signals accurately. The research discusses and evaluates the techniques for robust feature extraction, handling non-stationarity and spatio-spectral filter selection. The main objectives of the study could be summarized as follows:

- **Development of a Robust Feature Extraction Algorithm:** Development of a robust feature extraction algorithm capable of eliminating trials affected by artifacts and discarding redundant and noisy channels in order to improve the classification performance of the BCI system. The algorithm should be polynomial in time.
- **Development of an Algorithm to Handle Non-stationarity:** Development of an Interval Type-2 Neuro-Fuzzy Inference System (IT2NFIS) to handle non-stationarity in the EEG data. The algorithm should be able to handle the non-stationarity as uncertainty in classifier using interval type-2 fuzzy sets. The running time of the developed algorithm should be polynomial in time.

- **Self-regulation in Learning EEG Trials:** Incorporate self-regulation in the learning mechanism such that the parameters of the algorithm are adapted based on the novelty of the training sample. Samples with novel information content should be learnt first and the ones with low information content should be used to fine-tune the network.
  - **Computationally Efficient Type-Reduction:** It is intended to develop a computationally efficient type-reduction approach that does not use the expensive Karnik-Mendel (K-M) algorithm [27]. Development of type-reduction technique of this kind will render the inference mechanism faster.
  - **Online/Sequential Learning Framework:** To develop a learning algorithm for an IT2NFIS in a sequential learning framework, such that the network grows, updates and prunes the rules to achieve a parsimonious network. This network should be able to handle samples arriving one-by-one and also temporally varying data.
  - **Adaptive Framework:** To develop a learning algorithm that is able to adapt the network for gradual changes and also evolve it for sudden changes.
- **Development of a Mechanism to Find Subject-specific Spatio-spectral Filters:** To develop a system that finds spatio-spectral filters for a particular subject in the presence of non-stationarity in EEG signals. The developed mechanism should also provide a good classification performance. The spatio-spectral filter selection algorithm should have a polynomial running time.
  - **Realization of a Real World BCI system:** To develop a mechanism to control a quadcopter using the motor-imagery signals.

1.4 Major Contributions of the Thesis

The major contributions of this thesis are given below:

- **A Robust Common Spatial Pattern Feature Extraction Algorithm:** In this thesis, a Robust Common Spatial Pattern (RoCSP) feature extraction algorithm is proposed that eliminates trials affected by artifacts and discards redundant

channels in order to improve the classification performance of the BCI system. This algorithm eliminates the channels that do not carry distinguishable information for any two different motor-imagery tasks. It employs z -scores to eliminate noisy trials for a channel, followed by an effect-size calculation to eliminate the redundant channels. The results indicate that the proposed RoCSP algorithm outperforms other state-of-the-art feature extraction algorithms in the literature.

- **Self-regulated Interval Type-2 Neuro-Fuzzy Inference System:** A Self-Regulated Interval Type-2 Neuro-Fuzzy Inference System (SRIT2NFIS) is proposed to handle non-stationarity. It handles the non-stationarity in the input features by modeling it as uncertainty in the classifier. A five-layered modified Takagi-Sugeno-Kang interval type-2 fuzzy inference mechanism forms the structure and the learning algorithm is a self-regulatory learning mechanism. It employs a regularized projection based learning to compute the optimal weights of the network to avoid over-fitting. The experimental studies clearly show that the proposed algorithm outperforms other state-of-the-art type-1 and type-2 algorithms.
  - **Self-regulating Sequential Learning Algorithm:** The samples arrive sequentially and one-at-a-time and the learning algorithm decides whether a sample should be used for rule growing/parameter update or discarded without learning or reserved to be considered for learning at a later stage. It starts with zero rules and as a sample is presented to the network, the learning algorithm monitors the knowledge in the sample and decides whether to delete, learn or keep the sample for future use.
  - **Computationally Efficient Type-Reduction:** A computationally efficient type-reduction algorithm is introduced that employs a Nie-Tan type-reduction technique [28], proven to provide a closed form approximation of the well-known K-M algorithm. The use of this type-reduction technique renders the inference mechanism faster.
  - **Regularization of the Projection Based Learning Algorithm:** The projection based learning algorithm [29] computes the weights of the network

by minimizing the sum of squared hinge-error. However, it is prone to over-fitting. In this thesis, the sum of squared hinge-error is regularized in order to avoid over-fitting.

- **Intelligent-BCI System for Handling Non-stationarity in EEG Data:** An Intelligent-BCI (I-BCI) system is proposed by combining the RoCSP and the SRIT2NFIS algorithm for handling non-stationarity in EEG data. Here, the SRIT2NFIS classifier uses the input features generated by the RoCSP algorithm that are non-stationary in nature and handles them as uncertainty using interval type-2 fuzzy sets in the antecedent of rules. The proposed system is able to handle intra/intersession non-stationarity.
- **Subject-specific Spatio-spectral Filter Selection:** An approach to find the optimal subject-specific spatio-spectral filters for motor-imagery based BCI is proposed. The spatio-spectral filter selection is cast into the problem of band selection in a filter bank, followed by spatial filter selection in the selected bands. In the first step, SRIT2NFIS is used to eliminate unnecessary bands from a filter bank to arrive at a set of optimal spectral filters, referred to as Spectral-I-BCI (S-IBCI) method. This is an iterative method that eliminates those bands from the filter bank that do not affect the SRIT2NFIS training accuracy when dropped from consideration. In the second step, SRIT2NFIS uses those RoCSP filters from the selected bands in the first step and eliminates unnecessary spatial filters to obtain an optimal set of spatial filters. At the end of the second step, the proposed method referred to as Spatio-Spectral-I-BCI (SS-IBCI), is able to identify the desired spatio-spectral filters for a given subject. The experimental results indicate that, both, the S-IBCI and SS-IBCI methods perform better than state-of-the-art methods in the literature.
- **Realization of I-BCI to Control a Quadcopter:** In this thesis, the realization of I-BCI for control of quadcopter movement is presented. The device used to record raw EEG data in this work is a fourteen channel EMOTIV Epoc and AR.drone 2.0 is used as the control device. A framework is developed to convert the raw

EEG signals into commands to control the quadcopter. The RoCSP algorithm is used for feature extraction and the SRIT2NFIS classifier is employed to classify the signals into left/right classes. The decoded intentions of the user commands (move left, move right) are relayed to the quadcopter over a wireless connection. The experiments show that the user is able to control the quadcopter successfully.

1.5 Thesis Organization

This thesis is organized as follows:

- In **Chapter 2**, existing brain signal processing algorithms have been reviewed. They have been categorized as follows: temporal domain based approaches, frequency domain based approaches, time-frequency based approaches, spatial filtering approaches, and spectral filtering approaches.
- **Chapter 3** presents a Robust Common Spatial Pattern (RoCSP) feature extraction algorithm that eliminates the artifact affected trials and redundant channels to improve the accuracy of the BCI system. Performance comparison of RoCSP algorithm with other feature extraction algorithms shows improved performance on benchmark data sets.
- In **Chapter 4**, a Self-Regulated Interval Type-2 Neuro-Fuzzy Inference System (SRIT2NFIS) is proposed for handling non-stationary and uncertain data. The structure of SRIT2NFIS is a five-layered Takagi-Sugeno-Kang interval type-2 fuzzy inference mechanism. A self-regulatory learning mechanism is used to evolve the structure automatically and learn the parameters of the network. Performance comparison of SRIT2NFIS with other type-1 and type-2 systems shows better performance of the proposed algorithm.
- **Chapter 5** addresses the key problem of non-stationarity in EEG based BCI systems. This chapter proposes an Intelligent-Brain Computer Interface (I-BCI) system that handles, both, intra/intersession non-stationarity in EEG data. It uses the features generated by RoCSP algorithm and employs SRIT2NFIS for classification of EEG data. Using benchmark data sets, the results indicate that I-BCI

produces a higher classification accuracy in case of intra/intersession data set, as compared to other existing algorithms.

- In **Chapter 6**, the problem of finding the optimal subject-specific spatio-spectral filters for motor-imagery based BCI is addressed. Two methods, Spectral-I-BCI (S-IBCI) and Spatio-Spectral-I-BCI (SS-IBCI) are proposed for finding spectral and spatial filters. Performance comparison with existing methods in the literature shows superior performance of the proposed methods.
- **Chapter 7** presents a realization of I-BCI for control of quadcopter movement. Results show that the user is able to control the movement of the quadcopter.
- **Chapter 8** summarizes the conclusions from the study and presents directions for future works.

Chapter 2

A Review of Brain Signal Processing Algorithms for Motor-imagery based Brain Computer Interfaces

Development of Brain Computer Interface (BCI) system is an active research area with great benefits to the society. The main objective in developing a BCI system is to translate the neural activity generated by the brain into user commands. This thesis focuses primarily on motor-imagery based BCI systems where the user performs mental tasks to control the external devices without performing any muscle activity. In a majority of current motor-imagery based BCI systems, brain signals are measured using an Electroencephalography (EEG). During mental imagination of different motor tasks, event related de-synchronization/ synchronization patterns [16, 30] are generated in the EEG. EEG signal processing is performed to extract meaningful features in order to accurately classify the performed motor-imagery tasks. Hence, EEG signal processing is an important step in the development of an effective BCI. In this chapter, a review of existing EEG signal processing algorithms has been provided. The EEG signal processing algorithms proposed in the literature can be categorized into frequency domain based approaches, temporal domain based approaches, time-frequency based approaches, spatial filtering approaches and spectral filtering approaches as shown in Fig. 2.1.

Figure 2.1: Categorization of brain signal processing algorithms.

2.1 Frequency Domain Based Approach

In the literature, various frequency domain based signal processing approaches have been proposed for motor-imagery classification [31–36]. In [31], the power of frequency bands in which ERD/ERS patterns occur has been used as features to classify motor-imagery tasks. In [32–34], the band power estimates have been used to control external devices. The power spectral density and complex band power features have also been used for feature extraction [35, 36]. The main limitation of frequency domain approaches is that they require prior knowledge of subject-specific frequency bands.

2.2 Time Domain Based Approach

The signal processing approaches in time domain use the temporal variation in EEG signals as features. Time domain parameters have been used as features in EEG based BCI systems [37, 38]. In [7, 39], auto-regressive parameters have been used as features. The amplitude of the raw EEG signals [40, 41] and Hjorth parameters [7] have also been used as time domain features in EEG based BCI systems. These methods are simple in nature and do not consider the frequency component in which ERD/ERS patterns are generated.

2.3 Time-frequency Based Approach

In the literature, EEG signal processing approaches have been proposed that combine the temporal and frequency domain approaches to obtain time-frequency features [42–46]. The methods [42, 43] were based on narrowband frequency decomposition, and classification on either temporal features weighted over frequencies and sensor locations [43] or spatial patterns synthesized over time and frequency [42]. In [44], these methods were refined for the purpose of performing online analysis and tested by analyzing EEG data collected during online one-dimensional "cursor control" BCI experiments on two subjects. Short time Fourier transform has been also used in the literature to extract features from the EEG signals generated by right/left-hand imagery [45]. In [46], a subject-based feature extraction method using wavelet packet basis decomposition is proposed. It employs the wavelet packet basis algorithm to adapt to each subject separately. The EEG signals are decomposed to a given level by the wavelet packet transform followed by the computation of the best-adapted basis for a particular subject. Subband energies contained in the best basis are used as effective features. The limitation of purely using time-frequency features is that they assume that the EEG signal is stationary in nature.

2.4 Spatial Filtering Approach

Raw EEG scalp potentials are known to have a poor spatial resolution owing to volume conduction [47]. Spatial filtering is extremely useful in improving the signal-to-noise ratio and provide discriminable features. These spatial filters provide a linear combination of the raw EEG signals measured at different locations of the brain. The filters are computed such that the EEG signals corresponding to the different classes are projected onto a common orthogonal space. In the literature, many spatial filtering algorithms have been proposed [17, 48, 49]. These algorithms aim to find the spatial information present in the EEG signals. The Common Spatial Pattern (CSP) [17, 50, 51] is one the popular spatial filtering algorithm used in the literature. It is based on the decomposition of the EEG signals into spatial patterns such that the difference between the classes is maximized.

2.4.1 Common Spatial Pattern Algorithm

The Common Spatial Pattern (CSP) algorithm is highly successful in computing spatial filters for detecting the effect of ERD/ ERS patterns with respect to their topographic patterns. Early works on applying the CSP algorithm to extract abnormalities present in an EEG were presented in [52, 53]. Further, CSP was used to discriminate movement-related patterns [54]. Recently, CSP has been used in BCI domain with great success [5, 35]. CSP algorithm distinguishes the two classes of EEG data by minimizing the variance of one class of EEG data while maximizing the variance of the other class. It employs simultaneous diagonalization of two class covariance matrices to compute the spatial filters. The algorithm to compute the spatial filters is described below in detail.

For analysis, let us assume that the band passed EEG data of j -th trial and c -th class label is represented as $\mathbf{E}_c^j \in \mathbb{R}^{M \times S}$, where M represents the total number of channels and S is the number of measurement samples in each channel. The class-specific covariance matrix $\mathbf{C}_c^j$ of the EEG data of j -th trial is given as:

$$\mathbf{C}_c^j = \frac{\mathbf{E}_c^j \mathbf{E}_c^{jT}}{\text{trace}(\mathbf{E}_c^j \mathbf{E}_c^{jT})}; \quad c = 1, 2. \quad (2.1)$$

where, T is the transpose operator and $\text{trace}(\cdot)$ denotes the sum of the diagonal elements. The covariance matrix of the classes, $\mathbf{C}_1$ and $\mathbf{C}_2$ are calculated by taking an average over all the trials of EEG data of the respective classes.

The composite covariance matrix $\mathbf{C}_L$ is given as:

$$\mathbf{C}_L = \mathbf{C}_1 + \mathbf{C}_2 \quad (2.2)$$

On the covariance matrix $\mathbf{C}_L$, an eigen decomposition is performed and is given as:

$$\mathbf{C}_L = \mathbf{H}_c \chi \mathbf{H}_c^T \quad (2.3)$$

where, $\mathbf{H}_c$ denotes the matrix of normalized eigenvectors with corresponding eigenvalues χ .

The whitening transformation matrix is given as:

$$\mathbf{P} = \sqrt{\chi^{-1}} \mathbf{H}_c^T. \quad (2.4)$$

It decorrelates the composite covariance matrix $\mathbf{C}_L$ as given below:

$$\mathbf{P}\mathbf{C}_L\mathbf{P}^T = \mathbf{P}(\mathbf{C}_1 + \mathbf{C}_2)\mathbf{P}^T = \mathbf{I} \quad (2.5)$$

$$\mathbf{C}_1' = \mathbf{P}\mathbf{C}_1\mathbf{P}^T, \quad \mathbf{C}_2' = \mathbf{P}\mathbf{C}_2\mathbf{P}^T \quad (2.6)$$

Now, the common eigenvectors $\mathbf{U}$ for matrices $\mathbf{C}_1'$ and $\mathbf{C}_2'$ are found with corresponding eigenvalues summing to one,

$$\mathbf{C}_1' = \mathbf{U}\beta_1\mathbf{U}^T, \quad \mathbf{C}_2' = \mathbf{U}\beta_2\mathbf{U}^T, \quad \beta_1 + \beta_2 = \mathbf{I} \quad (2.7)$$

where, $\mathbf{I}$ is the identity matrix. Here, the eigenvector with the largest eigenvalue for $\mathbf{C}_1'$ has the smallest eigenvalue for $\mathbf{C}_2'$ and vice-versa.

The projection matrix $\mathbf{V} = (\mathbf{U}^T\mathbf{P})$ are called spatial filters. From $\mathbf{V}$, as per the requirement, different settings of most dominant pair of spatial filters $\mathbf{V}'$ for task 1 and task 2 are chosen, respectively. The projected signal from the selected spatial filters can be used to map a trial $\mathbf{E}$ given by:

$$\mathbf{Z} = \mathbf{V}' \times \mathbf{E}. \quad (2.8)$$

The transformed signals, $\mathbf{Z}$ maximize the variance between the two classes. The log-variances of the signals are computed as features and are given by:

$$\mathbf{x}_j = \log(\text{var}(\mathbf{Z})); \quad j = 1, 2, \dots, 2m \quad (2.9)$$

where, $\mathbf{x}_j = [x_{j1}, \dots, x_{j2m}]^T \in \mathbb{R}^{2m}$ is a $2m$ -dimensional feature vector. These spatial features are the log-variances of the projected EEG signal from the selected spatial filters. The spatial filters provide a linear combination of the raw EEG signals measured at different locations of the brain. These filters are computed such that the EEG signals are projected onto an orthogonal space. Equations (2.8) and (2.9) are repeated for every trial to compute the log-variance features. CSP has been employed to extract features which are then used to perform classification [55–57]. Different variations of CSP have been proposed in the literature. In [58–60], CSP has been used in a predefined region of interest. Some CSP variants consider the temporal structure of the samples [61–63]. CSP has also been extended for multiclass classification using joint approximate

diagonalization [64, 65], information theory [66] and Kullback-Leibler divergence [67]. Wavelet CSP methods have also been proposed in the literature in [68, 69]. In [70], CSP has been improved by considering both phase and amplitude of EEG signals.

However, CSP is known to be highly sensitive to variability and non-stationarity which is inherent to EEG data [71, 72]. Many algorithms have been presented in the literature that modify the CSP algorithm to handle the non-stationarity in EEG data. They can be categorized into algorithms that adapt the CSP features [73, 74] and algorithms that regularize the CSP algorithm [71, 75–78]. Next, a review on adaptive as well as regularized CSP algorithm is presented.

2.4.2 Adaptive CSP Algorithms

In the literature, various algorithms have been proposed that adapt the CSP algorithm to handle the non-stationarity in EEG data [73, 74, 79, 80]. In [73], an expectation minimization approach has been presented that re-extracts the features using CSP based on the samples from test data in an iterative manner. Here, feature extraction using the CSP algorithm and the classification of CSP features are performed iteratively. In each iteration, the training data set is updated using samples from the test data set along with the predicted labels. The CSP features are recomputed based on the new training data and classified using the standard expectation-minimization algorithm. In adaptive common spatial pattern [74], the covariance matrix has been adapted for every new segment of EEG data. Here, whenever a new trial or EEG segment for test session is encountered, the covariance matrix is updated as the weighted sum of historical covariance and the covariance of the current recording segment. After updating the covariance matrix, the standard CSP procedure is employed to compute the spatial filters. However, if the raw signal is non-stationary, adapting the CSP features will only account for small changes. In [79, 80], data from evaluation section has been used to adapt the CSP algorithm to handle non-stationarity and in [81, 82], data from previously recorded session has been used to handle the non-stationarity. In [25, 83], samples from testing data have been used to adapt the trained model to handle the non-stationarity present in EEG data and in [84–86], approaches based on covariance shift adaptation have been presented.

2.4.3 Regularized CSP Algorithms

In the literature, various algorithms have been proposed for the regularization of the CSP algorithm to handle non-stationarity in EEG data. In this section, the algorithms that regularize the CSP algorithm are described in detail [71, 75–78].

- **Composite Common Spatial Pattern:**

In [75], two Composite CSP methods (CCSP1 & CCSP2) have been proposed that perform subject-to-subject transfer to regularize the CSP algorithm. These algorithms regularize the covariance estimate of a subject by adding a generic covariance matrix to it. In CCSP1, the generic covariance matrix is computed by assigning weights to the covariance matrices of other subjects and performing a weighted sum. In CCSP2, the weights to the covariance matrices of other subjects are assigned based on the Kullback-Leibler (KL) divergence between the subject's data and the target subject. The generic matrix is computed using the weighted sum of the covariance matrices of other subjects. The generic matrix G_c is given as:

$$G_c = \sum_{i \in \Omega} \frac{1}{\mathcal{Z}} \frac{1}{KL(i, t)} C_c^i \quad (2.10)$$

with

$$\mathcal{Z} = \sum_{i \in \Omega} \frac{1}{KL(i, t)} \quad (2.11)$$

Here, Ω denotes the set of subjects, $KL(i, t)$ is the KL-divergence between subject i and the target subject t and is given as:

$$KL(i, t) = \frac{1}{2} \left(\log \left(\frac{\det(C_c)}{\det(C_c^i)} \right) + \text{tr}(C_c^{-1} C_c^i) - N_e \right) \quad (2.12)$$

where, $\det$ and tr are, respectively, the determinant and the trace of a matrix, and N_e is the number of electrodes used. $\mathcal{Z}$ represents the inverse Kullback-Leibler divergence and C_c^i denotes the spatial covariance matrix for class c and subject i .

- **Regularized Common Spatial Pattern with Generic Learning:**

In Regularized CSP with Generic Learning (GLRCSP) [76], the covariance matrix has been regularized using data from other subjects aiming to shift it towards an identity matrix as well as a generic covariance matrix. The covariance matrix is given as:

$$\tilde{C}_c = (1 - \gamma)\hat{C}_c + \gamma I \quad (2.13)$$

with

$$\hat{C}_c = (1 - \beta)s_c C_c + \beta G_c \quad (2.14)$$

Here, C_c is the initial spatial covariance matrix for class c , $\tilde{C}_c$ is the regularized estimate, I is the identity matrix, s_c is a scaling constant and β and γ are regularization parameters.

- **Regularized Common Spatial Pattern with Diagonal Loading:**

Regularized CSP with Diagonal Loading (DLCSP) [71] also regularized the covariance matrix aiming to shift it towards the identity matrix. It has three variants: DLCSPauto, DLCSPcv and DLCSPcvdiff. In DLCSPauto, the regularization parameter is automatically determined using Ledoit and Wolf method (one for class ‘one’ covariance matrix and another for class ‘two’ covariance matrix) [87]. DLCSPcv uses cross-validation to select a single regularization parameter for covariance matrices of both the classes while DLCSPcvdiff uses cross-validation to select two different regularization parameters for the two classes.

- **Invariant Common Spatial Pattern:**

In Invariant CSP (ICSP) [77], the CSP objective function has been regularized with an aim to obtain filters that are invariant to a given noise source. The regularization matrix is defined as the covariance matrix of the noise source (e.g., covariance matrix of the varying level of occipital α -activity). The covariance matrix of the noise is incorporated in the denominator of the CSP objective function. However, this method required additional EEG measurements to obtain signals for the covariance matrix computation.

- **Regularized Common Spatial Pattern with Selected Subjects:**

In Regularized CSP with Selected Subjects (SSRCSP) [71, 88], similar to CCSP and GLRCSP, data from other subjects has been utilized to shrink the covariance matrix towards a generic matrix. However, SSRCSP utilizes data from selected number of subjects instead of all the subjects due to the presence of large intersubject variation. The generic covariance matrix G_c for SSRCSP algorithm is defined as:

$$G_c = \frac{1}{|S_t(\Omega)|} \sum_{i \in S_t(\Omega)} C_c^i \quad (2.15)$$

where, $|B|$ is the number of elements in set B and $S_t(\Omega)$ is the subset of selected subjects. To arrive at an appropriate set of subjects, SSRCSP selects the subjects sequentially to add or discard them from the current subset of subjects. The objective is to maximize the accuracy using the training data from a particular subject and testing data of the target subject.

- **Common Spatial Pattern with Tikhonov Regularization:**

In CSP with Tikhonov Regularization (TRCSP) [71], the CSP objective function has been regularized with Tikhonov regularization (TR). TR is a form of regularization used for regression problems [89], which penalizes solutions with larger weights. The penalty term used here is given as:

$$P_l(v) = ||v||^2 = v^T v = v^T I v. \quad (2.16)$$

where, v is the spatial filter obtained using CSP. This penalty term mitigates the influence of artifacts and reduces the tendency to overfit. TRCSP has a regularization parameter which provides a trade-off between the maximization of the CSP objective function and the minimization of the penalty term.

- **Common Spatial Pattern with Weighted Tikhonov Regularization:**

In CSP with Weighted Tikhonov Regularization (WTRCSP) [71], the CSP algorithm has been regularized with a weighted TR which penalizes channels with different penalties. The intuition behind WTRCSP is that some channels are more important than others to classify a given motor-imagery task. Hence, WTRCSP

assigns different penalties for different channels. The channel that is less likely to have a significant contribution in the spatial filters is given a relatively larger penalty in order to prevent CSP from assigning it a large contribution. On the other hand, a channel that is likely to be useful is assigned lesser penalty. The penalty term used here is given as $P_l(v) = v^T D_v v$, where D_v is a diagonal matrix such that $D_v = \text{diag}(v_G)$ and $v_G(i)$ is the level of penalty assigned to channel i . Here, the penalty levels $v_G(i)$ is chosen using data from other subjects as:

$$v_G = \left(\frac{1}{2 \times N_f \times |\Omega|} \sum_{i \in \Omega} \sum_{f=1}^{2 \times N_f} \left| \frac{v_f^i}{\|v_f^i\|} \right| \right)^{-1} \quad (2.17)$$

where, v_f^i is the f -th spatial filter obtained using CSP for the i -th channel, Ω denotes the set of subjects. The penalty level of a channel is set proportional to the inverse of its importance in the obtained CSP filters.

- **Spatially Regularized Common Spatial Pattern:**

Spatially Regularized CSP (SRCSP) [71, 90] considered the spatial location of EEG electrodes and obtains spatially smooth filters for which the neighboring electrodes have relatively similar weights. SRCSP leverages on the fact that neighboring neurons have similar functionality, hence, neighboring electrodes should measure similar brain activity (electrodes that are closer to each other). To ensure spatial smoothness of the filters v , SRCSP employs a Laplacian penalty term $P_l(v)$ with the regularization matrix given as:

$$R_m = D_G - G \quad (2.18)$$

where

$$G(i, j) = \exp \left(-\frac{1}{2} \frac{\|\rho_i - \rho_j\|^2}{r^2} \right) \quad (2.19)$$

Here, D_G is a diagonal matrix such that $D_G(i, i) = \sum_j G(i, j)$, ρ_i denotes the vector with the i th electrode coordinates, G is a Gaussian kernel and r is a parameter that controls the distance between two electrodes. The penalty term is large for filters in which neighboring electrodes have very different weights and is given as:

$$P_l(v) = v^T (D_G - G) v = \sum_{i,j} G(i, j) (v_i - v_j)^2. \quad (2.20)$$

- **Kullback-Leibler Common Spatial Pattern:**

Kullback-Leibler CSP (KLCSP) [91] maximized the discrimination between the mean of two classes while minimizing the within-class distance. The KL divergence based term is defined to measure the within-class dissimilarities. Variability is measured as the average Kullback-Leibler divergence between the data distribution of a trial and the average distribution of the corresponding class. Here, the following objective function is optimized:

$$\min_v \left((1 - \nu) v^T C_c v + \nu \frac{1}{2N} \sum_{c=1}^2 \sum_{i=1}^N D_{kl}(v^T C_c^i v || v^T C_c v) \right) \quad (2.21)$$

subject to:

$$V^T (C_1 + C_2) V = 1 \quad (2.22)$$

where, C_c^i is the covariance matrix estimated from the i -th trial of class c and C_c is the average class covariance matrix. The parameter ν is trade-off between discrimination ability and stationarity of data.

- **Stationary Common Spatial Pattern:**

The stationary CSP (sCSP) algorithm aimed to obtain filters that maximize or minimize variances for two different conditions, but at the same time maintain the variance estimation along the projected direction as stable as possible across trials. It aims to regularize CSP towards stationary subspaces such that it becomes invariant to non-stationarities in EEG data.

2.4.4 Summary of Spatial Filtering Approaches

In this section, a review of spatial filtering algorithms has been presented. CSP is an effective feature extraction approach commonly used in this area that computes spatial filters from the raw EEG data, such that the separation between two classes is maximized. In order to handle the inherent non-stationarity present in EEG data, CSP has been regularized or adapted using the test samples. A summary of the presented algorithms has been provided in Table 2.1. The frequency ranges on which these algorithms

operate are either chosen manually or a broad frequency range is used. However, these frequency ranges are subject-specific i.e. different subjects respond to a different range of frequencies. The choice of frequency bands also affect the corresponding spatial filters. Hence, there is a need to find the subject-specific frequency ranges (spectral filters) and their corresponding spatial filters for accurate motor-imagery task classification. Next, a detailed review is provided on algorithms that aim to find subject-specific spectral and spatial filters.

2.5 Spectral Filtering Approaches

During mental imagination of different motor tasks, event related de-synchronization/synchronization patterns are generated in the EEG data. Event-Related De-synchronization (ERD) pattern (short-lasting attenuation or blocking of rhythms) occurs in the (μ (8-12) Hz and beta(14-30) Hz) frequency bands. The opposite phenomenon of ERD is called Event-Related Synchronization (ERS) pattern, where an increase in spectral peak occurs in μ and beta bands. These patterns are useful phenomena in motor-imagery tasks for accurate detection of the task. For more details on ERD, and ERS, refer to [16, 30]. These ERD/ERS patterns occur in different frequency ranges for different subjects [92]. Based on these subject-specific patterns, different spatial locations of the brain have different activation which makes feature extraction from a specific location challenging. Hence, there is a need to find the subject-specific frequency ranges (using band-pass filters referred to as spectral filters) followed by spatial filters for accurate motor-imagery task classification. In the literature, various algorithms have been developed for addressing the problem of selecting the subject-specific spectral and spatial filters.

- **Common Spatio-Spectral Pattern:**

In the Common Spatio-Spectral Pattern (CSSP) method [49], the CSP algorithm has been extended by appending a time-delayed EEG signal as the signal from additional channels resulting in additional finite impulse response filters at each electrode. In order to extract features, Eqn. 2.8 is extended by one delayed coordinate to form the following equation

$$Z = V^0 E + V^\tau \zeta^\tau E \quad (2.23)$$

Table 2.1: Summary of major spatial filtering algorithms discussed in literature review.

Algorithm	Description	Artifacts/ Non-stationarity
CCSP1, Kang et al. [75]	Regularizes the class covariance matrix of a subject towards a generic covariance matrix constructed using data of other subjects.	Partially
CCSP2, Kang et al. [75]	Regularizes the class covariance matrix of a subject by constructing the generic covariance matrix using weighted Kullback-Leibler divergence.	Partially
GLRCSP, Lu et al. [76]	Covariance matrix is regularized using data from other subjects aiming to shift it towards an identity matrix as well as a generic covariance matrix.	Partially
DLCSPauto, Lotte and Guan [71]	Regularizes the covariance matrix aiming towards the identity matrix by choosing separate regularization parameter for two classes using Ledoit and Wolf method [87].	Partially
DLCSPcv, Lotte and Guan [71]	Regularizes the covariance matrix aiming towards the identity matrix by selecting a single regularization parameter using cross-validation for covariance matrices of both the classes.	Partially
DLCSPcvdiff, Lotte and Guan [71]	Regularizes the covariance matrix aiming towards the identity matrix by using cross-validation to select two different regularization parameters for the two classes.	Partially
ICSP, Blankertz et al. [77]	Regularizes the CSP objective function to obtain filters that are invariant to a given noise source. Regularization matrix is defined as the covariance matrix of the noise source.	Partially
TRCSP, Lotte and Guan [71]	CSP objective function has been regularized with Tikhonov regularization.	Partially
WTRCSP, Lotte and Guan [71]	CSP algorithm has been regularized with a weighted Tikhonov regularization that penalizes channels with different penalties.	Partially
SRCSP, Lotte and Guan [90]	Spatial location of EEG electrodes is considered and spatially smooth filters are obtained for which the neighboring electrodes have relatively similar weights.	Partially
KLCSP, Arvaneh et al. [91]	Adds KL divergence regularization term to CSP in order to extract stationary features.	Partially

where, $\zeta^\tau E$ denotes the signal E delayed by time τ , V^0 and V^τ represent the projection matrices related to E and $\zeta^\tau E$ respectively. Appending the delayed vectors $\zeta^\tau E$ as additional channels to E , the new EEG matrix $\hat{E}$ is obtained as:

$$\hat{E} = \begin{pmatrix} E \\ \zeta^\tau E \end{pmatrix} \quad (2.24)$$

The optimization criterion is to find projections of V^0 and V^τ such that the signal variances of different Z best discriminates the two given classes, i.e. maximize the variance for one class while minimizing it for the opposite class. The Equations 2.23 and 2.24 are re-arranged as:

$$\hat{V}\hat{E} = \left(\hat{V}^0\hat{V}^\tau\right)\hat{E}. \quad (2.25)$$

The optimum solution for the decomposition matrix $\hat{V}$ includes the interpretations of both spatial and spectral filters. In CSSP, the original CSP approach is applied to the concatenation of E and $\zeta^\tau E$ in the channel dimension. The delayed signals $\zeta^\tau E$ are treated as new channels and therefore, its computational demands are doubled compared to the original CSP.

- **Common Sparse Spectral Spatial Pattern:**

In the CSSP algorithm, the frequency filters and the spatial filters have been optimized simultaneously. However, the CSSP algorithm is not flexible due to the use of a single time-delay filter. Hence, in Common Sparse Spectral Spatial Pattern (CSSSP) [93], the authors extended the CSSP algorithm by allowing a larger time-delay. It optimizes an arbitrary FIR filter within the CSP algorithm. The CSSSP algorithm performs better than both CSP and CSSP algorithm. However, the selection of time-delay parameter in CSSSP algorithm is computationally expensive.

- **Spectrally-Weighted Common Spatial Pattern:**

In Spectrally-weighted CSP (SPEC-CSP) algorithm [94], the spatial projection matrix and the coefficients of cross-spectrum matrices have been optimized using two different cost functions. It starts from the conventional CSP method and alternately updates the spectral weighting coefficients and the spatial projection. The algorithm is run for two iterations since two iterations provide good classification performance. Here, Fisher's criterion with positive coefficient constraint is used to design the spectral filters.

- **Iterative Spatio-Spectral Pattern Learning:**

In Iterative Spatio-Spectral Patterns Learning algorithm (ISSPL) [95], the spatial and spectral filters have been learned automatically using statistical learning theory. It simultaneously parameterizes the spectral filters along with the classifier to achieve good classification performance. It learns the spatio-spectral filters and the classifier from the data in a sequential and iterative manner. Here, contrary to the sequence of spectral filtering and spatial filtering, first the spatial filtering is performed on the EEG signals followed by the tuning of temporal filters based on the learned spatial filters. The alternating optimization algorithm is repeated until a stopping criterion is achieved.

- **Sub-Band Common Spatial Pattern:**

In Sub-Band Common Spatial Pattern (SBCSP) [96], a broad frequency range has been divided into subbands and a reduced set of features was selected. A score for each subband was computed using the Linear Discriminant Analysis (LDA) and the top bands with higher scores were selected. The filter bank uses Gabor filters of bandwidth 4 *Hz* in range of 8-40 *Hz*. The spatial filters that use CSP algorithm are then employed on each of these subbands. After obtaining subband scores using a LDA, a recursive band elimination or a classification algorithm is employed to select the subbands as shown in the schematic diagram in Fig. 2.2.

- **Filter Bank Common Spatial Pattern:**

Filter Bank CSP (FBCSP) [92] employed a mutual information based feature selection approach to select a subset of features generated by a filter bank followed by the band selection. FBCSP comprises of 4 stages: frequency filtering, spatial filtering, feature selection, and classification as shown in Fig. 2.3. The first stage employs a filter bank of nine bandpass filters of bandwidth 4 *Hz* to split the multichannel EEG signals (4-8 *Hz*, 8-12 *Hz*, 12-16 *Hz*, 16-20 *Hz*, 20-24 *Hz*, 24-28 *Hz*, 28-32 *Hz*, 32-36 *Hz* and 36-40 *Hz*). The second stage performs spatial filtering on each of these bands using the CSP algorithm and extracts CSP features. In the third stage, a feature selection algorithm is employed to select the discriminative frequency bands and CSP features. In the fourth stage, a classification algorithm is used to classify the selected CSP features to obtain the class label.

Figure 2.2: Schematic diagram of SBCSP.

Figure 2.3: Schematic diagram of FBCSP.

- **Discriminative Filter Bank Common Spatial Pattern:**

The Discriminative Filter Bank CSP (DFBCSP) [97] algorithm employed a filter bank selection before feature extraction. It uses subject-specific Discriminative Filter bank (DFB) instead of using a fixed filter bank for all subjects. The parent filter bank is designed using a coefficient decimation technique and it covers all the frequency components in the range of 6-40 Hz . The EEG channels $C3$ and $C4$ are selected in order to determine the subject-specific discriminative frequency components as shown in the schematic diagram in Fig. 2.4. In the band selection procedure, the parent filter bank filters EEG from $C3$ or $C4$ and the Fisher ratio of filtered EEG is used to determine the subject-specific discriminative frequency bands. After the selection of subject-specific frequency bands, the multi-channel EEG data is filtered using these discriminative bands and applied to the CSP algorithm for further processing. The Support Vector Machine (SVM) classifier is used for prediction of the output.

Figure 2.4: Schematic diagram of DFBCSP.

- **Optimal Spatio-Spectral Filter Network:**

In the Optimal Spatio-Spectral Filter Network (OSSFN) [98], an information theory based approach has been employed to optimize the spatio-spectral filters. It uses the mutual information between the features and class labels to optimize the filters. The broad frequency band is decomposed into an array of small bands and then spatial filter optimization is carried out using the CSP algorithm in each of these bands. Finally, the bands with higher mutual information are selected.

- **Maximum Mutual Information Spatial Spectral Features:**

In Maximum Mutual Information Spatial Spectral Features (MMISS) [99], the spatio-spectral filters have been optimized simultaneously. The learning of spatial and spectral filters are formulated as a mutual information maximization problem. The spectral and spatial filters are optimized by fixing one filter and optimizing the other and vice-versa. This procedure is conducted iteratively by employing an iterative gradient method to maximize the mutual information. It differs from OSSFN as it optimizes the spectral and spatial filters simultaneously, while OSSFN employs a filter bank based strategy. Also, it treats spatial and spectral filters as multivariate features following Gaussian distribution which makes the computation fast and efficient.

- **Bayesian Spatio-Spectral Filter Optimization:**

A spatio-spectral filter selection framework for extracting discriminative features has been presented in Bayesian Spatio-Spectral Filter Optimization (BSSFO) [100]. It assigns different weights to different frequency bands and employs a wrapper around method based on Bayesian framework for optimizing the spatio-spectral filters. Here, a Bayesian framework has been built which probabilistically selects the discriminative frequency bands and the corresponding spatial filters have been optimized. The problem of optimizing the spatio-spectral filters has been formulated as the estimation of the posterior probability density function that represents the relative probabilities of different states in the discriminating single-trial EEG. It combines the outputs from multiple classifiers using the weights of different frequency bands to construct a spectrally weighted label decision rule.

- **Adaptively Weighted Spectral-Spatial Pattern:**

In Adaptively Weighted Spectral-Spatial Pattern (AWSSP) [101], the discriminative frequency bands have been estimated using the Fisher criterion. It investigates the variability of frequency bands across different sessions of EEG data and tracks the variation in the frequency bands over time.

2.5.1 Summary of Spectral Filtering Approaches

In this section, a review of spectral filtering algorithms has been presented. A summary of the presented algorithms has been provided in Table 2.2.

Table 2.2: Summary of major spectral filtering algorithms discussed in literature review.

Algorithm	Description	Non-stationarity
CSSP, Lemm et al. [49]	CSP algorithm has been extended by appending a time-delayed EEG signal as the signal from additional channels resulting in additional finite impulse response filters at each electrode.	No
CSSSP, Dornhege et al. [93]	The CSSP algorithm [49] has been extended by allowing a larger time-delay.	No
SPEC-CSP, Tomioka et al. [94]	The spatial projection matrix and the coefficients of cross-spectrum matrices have been optimized iteratively.	No
FBCSP, Ang et al. [92]	A mutual information based feature selection approach is used to select a subset of features generated by a filter bank followed by the band selection.	No
DFBCSP, Thomas et al. [97]	Subject-specific discriminative filters selected based on Fisher ratio values are used to find spectral filters.	No
OSSFN, Zhang et al. [98]	An information theory based approach is employed to optimize the spatio-spectral filters.	No
BSSFO, Suk and Lee [100]	A wrapper around method based on Bayesian framework for optimizing the spatio-spectral filters.	No
AWSSP, Thomas et al. [101]	It investigates the variability of frequency bands across different sessions and tracks the variation in the frequency bands over time.	No

2.6 Summary

In this chapter, a review of brain signal processing algorithms has been presented. The algorithms proposed in the literature have been categorized into frequency domain based approaches, temporal domain based approaches, time-frequency based approaches, spatial filtering approaches and spectral filtering approaches. A short summary of the major spatial filtering approaches is provided in tabular form in Table 2.1 and spectral filtering approaches is provided in Table 2.2, respectively. Most of the above mentioned algorithms employ all the channels and use all the trials to learn the spatio-spectral filters. However, selection of all the channels and usage of complete set of trials may lead to inclusion of noisy channels and produce features that are not robust. In the next chapter, a robust feature extraction algorithm is presented. In subsequent chapters, the issues towards the development of an Intelligent BCI are addressed.

Chapter 3

A Robust Common Spatial Pattern Feature Extraction Algorithm

EEG based motor-imagery BCI systems translate the intentions of the subjects into user commands as shown in Fig. 3.1. EEG signals generated by the imagination of motor activities are recorded using multiple channels on a trial by trial basis in the brain activity acquisition phase. In the brain signal processing phase, features are extracted using the EEG signals followed by classification of EEG trials. A review of existing brain signal processing algorithms has been provided in Chapter 2. Feature extraction is an important step in brain signal processing for computing meaningful features that have the capability to distinguish the motor-imagery tasks. Most of the existing feature extraction algorithms consider all the trials and channels for feature extraction. However, considering all the trials might lead to a poor estimation of the class covariance matrices [19–22, 102–104]. Furthermore, employing a large number of channels may cause redundancy of data and might include noisy channels [105–112]. Also, using a large number of channels results in longer preparation time, which may be inconvenient for the subject using the BCI system. One of the challenge is to remove artifacts and eliminate redundant channels for an effective BCI.

This thesis aims to develop a unified BCI system that addresses the below mentioned challenges.

- Handle artifacts and eliminate redundant channels.
- Handle non-stationarity in EEG data.

- Subject-specific spatio-spectral filter selection.

Figure 3.1: Typical architecture of a motor-imagery based BCI system.

Figure 3.2: Brain signal processing framework.

This chapter focuses on the development of a robust feature extraction algorithm (highlighted in Fig. 3.2). Here, a Robust Common Spatial Pattern (RoCSP) feature extraction algorithm is proposed that eliminates trials affected by artifacts and discards redundant channels to improve the classification performance of the BCI system. The basic idea of RoCSP is to first eliminate the trials that may lead to a noisy estimation of class-specific covariance matrix followed by removal of channels that do not carry distinguishable information for the two different motor-imagery tasks. It eliminates the noisy trials by employing z -score [113] for a particular channel followed by an effect-size

calculation [114] to determine the channels that do not carry distinct information for the two different motor-imagery tasks. First, the z -scores of all the trials for a particular channel are calculated to eliminate the noisy trials. The z -score computation measures the distances of trials for a channel from the trial mean. This allows for identification of trials that are far away from the sample mean making it possibly a noisy trial. The trials that are frequently far from sample mean, across all the channels are excluded from the calculation of the channel mean. Next, the effect-size d is calculated by taking the sample mean and standard deviation for a particular channel for the two motor-imagery tasks. It provides a measure of whether the particular channel carries significant information to distinguish the two tasks in question. The channels with low d value are eliminated. This work has been published in [115].

The performance of RoCSP is evaluated using two publicly available BCI competition data sets, data set-IIa (six classification tasks) from BCI competition IV [116] and the data set-IVa from BCI competition III [117]. The performance of RoCSP algorithm is compared with other algorithms that regularize the CSP, including (GLRCSP, CCSP1, CCSP2, DLCSPauto, DLCSPcv, DLCSPcvdiff, SSRCS, TRCS, WTRCS and SRCSP).

The rest of the chapter is organized as follows. In Section 3.1, an illustrative example for the trial and channel selection is presented. Section 3.2 presents the proposed RoCSP algorithm. The data set used in this thesis are described in Section 3.3. The performance evaluation is presented in Section 3.4 and Section 3.5 provides the summary of the chapter.

3.1 Illustrative Example for Trial/Channel Selection

In this section, the trial and channel selection are illustrated with help of an example shown in the schematic Fig. 3.3. For the purpose of understanding, let us consider four channels for motor-imagery tasks A and B. Let the data of a single trial in a particular channel be represented by $\mathbf{x} \in \mathbb{R}^{1 \times S}$, where S is the number of measurement samples in the channel. The dot-product $\mathbf{x}\mathbf{x}^T$ is represented by the blocks in a particular channel as shown in the figure. It provides the time wise variance of trial for a channel. The taller

blocks represent the noisy trials in a particular channel. The trials that are noisy across channels are eliminated. In case of the CSP algorithm, all the trials were averaged for a particular class which led to a poor estimation of the class covariance matrix due to the trials affected by artifacts [19–22, 102–104]. This problem is addressed in the RoCSP algorithm by eliminating such trials that are affected by the artifacts from participating in the class covariance matrix calculation while computing spatial filters. The removal of noisy trials is also helpful in channel selection as a channel is eliminated based on the mean of trials for the two tasks. Inclusion of these noisy trials may affect the sample mean which in turn may lead to poor performance. Once the noisy trials are eliminated, the trial mean and variance for the channels are calculated. The channels with a similar trial mean (shown in a circle in the Fig. 3.3) for the two tasks are eliminated as they do not carry discriminative information for the two tasks. The trial selection is performed using z -scores of all the trials in a particular channel and channel selection using *Cohen's d* effect-size [114]. In the next section, RoCSP algorithm is described that eliminates trials affected by artifacts and discards redundant channels.

3.2 Robust Common Spatial Pattern Algorithm

In this section, the proposed robust common spatial pattern algorithm is presented in detail. Let the total number of recorded trials and channels be denoted as N and M , respectively. The objective of the RoCSP algorithm is to eliminate the noisy trials and select L channels ($L \leq M$) which provide better discriminability across classes, resulting in robust features. The problem of finding robust features with reduced number of channels is tractable as the time complexity of RoCSP is in polynomial time. It aims to improve the classification performance of the CSP algorithm by eliminating trials affected by artifacts and removing the redundant channels in the computation of the log-variance features. It eliminates the noisy trials in each channel followed by an effect-size based channel selection.

In order to find the noisy trials in each channel, the z -scores of all the trials in a particular channel is computed. It helps in identifying the trials that are far away from the sample mean making them possible noisy trials. Let c_{ci}^j be the dot-product of the

Figure 3.3: Illustrative example for trial/channel selection. The blocks show the dot product of trials for each channel. The red ellipses represent the eliminated noisy trials. The blue ellipses represent the eliminated channels.

j -th trial in i -th channel belonging to c -th class. The z -scores [113] of all the trials in the i -th channel is given as:

$$z_{ci}^j = \frac{|c_{ci}^j - \bar{c}_{ci}|}{\sigma_c}; \quad \forall j \quad (3.1)$$

Here, $\bar{c}_{ci}$ and σ_c represents the trial mean and standard deviation for i -th channel and c -th task, respectively. Now, the possible noisy trials in the i -th channel are calculated as:

$$s_{ij}^c = \begin{cases} 1, & \text{if } z_{ci}^j > \alpha_t \\ 0, & \text{otherwise} \end{cases} \quad (3.2)$$

Here, α_t represents the threshold parameter above which the trials are suspected as noisy. The α_t value can be chosen based on cross-validation. A lower value of α_t (close to zero) will result in non-noisy trials being termed as noisy, whereas a higher value will lead to the inclusion of noisy trials leading to poor performance. It is chosen in the range $[0.50, 1.90]$.

The frequency (in %) $f_{y_c}^j$ for all trials across all the M channels are given as:

$$f_{y_c}^j = \frac{\sum_{i=1}^M s_{ij}^c}{M} \times 100; \quad \forall j \quad (3.3)$$

The trials for which the frequency is greater than a particular threshold ζ is given as:

$$K_{tc} = \{j : f_{y_c}^j > \zeta\}; \quad \forall j \quad (3.4)$$

Here, K_{tc} represents the set of noisy trials for which frequency is high. The value of ζ can be chosen by cross-validation. A higher value of ζ is chosen in case of smaller number of channels with discriminative information, whereas a lower value of ζ is chosen in case of a large number of channels. The suggested range for this threshold is $[10, 20]$.

The i -th channel mean $\bar{c}_{ci}$ for c -th task is calculated by averaging the non-noisy trials and is given as:

$$\bar{c}_{ci} = \frac{1}{|J_s^c|} \sum_{j \in J_s^c} c_{ci}^j \quad (3.5)$$

with

$$J_s^c = \{j : \forall j \notin K_{tc}\} \quad (3.6)$$

Here, J_s^c represents the set of non-noisy trials that are included in the computation of mean in i -th channel.

In order to select a channel, the effect-size d of the channel is computed for the two tasks. The effect-size, d for the channel across two tasks is given as:

$$d_i = \frac{|\bar{c}_{1i} - \bar{c}_{2i}|}{\sigma_p} \quad (3.7)$$

with

$$\sigma_p = \sqrt{\frac{(|J_s^1| - 1)\sigma_{1i}^j + (|J_s^2| - 1)\sigma_{2i}^j}{|J_s^1| + |J_s^2| - 2}}; \quad \forall j \in J_s^c \quad (3.8)$$

where, σ_p is the pooled standard deviation, σ_{ci}^j represents the standard deviation of c -th class across the selected trials corresponding to i -th channel and $|J_s^1|, |J_s^2|$ are the number of trials in the two channels.

The channel i is selected if corresponding d -value is greater than a particular threshold, δ indicating the presence of discriminative information for the two tasks.

$$L = \{i : d_i > \delta\}; \quad \forall i \quad (3.9)$$

Here, the set L represents the set of selected channels. The threshold δ can be chosen using cross-validation. A higher value of this threshold will lead to retention of redundant channels and a lower value will lead to the elimination of non-redundant channels. The recommended range for δ is $[0.3, 1.3]$. The rationale behind Eqn. 3.9 is that channels with a similar trial mean for the two tasks are not selected as they do not carry discriminative information for the two tasks. The RoCSP algorithm employs the CSP algorithm [17, 52, 53] with selected trials and reduced number of channels to extract the log-variance features.

Let the EEG data of the j -th trial and c -th class label be represented $\mathbf{E}_c^j \in \Re^{L \times S}$, where L denotes the selected channels and S is the number of measurement samples in each channel. The class covariance matrix $\mathbf{C}_c^j$ of the j -th trial is given as:

$$\mathbf{C}_c^j = \frac{\mathbf{E}_c^j \mathbf{E}_c^{jT}}{\text{trace}(\mathbf{E}_c^j \mathbf{E}_c^{jT})}; \quad c = 1, 2 \quad (3.10)$$

where, T is the transpose operator and $trace(.)$ denotes the sum of diagonal elements. The covariance matrix of each of the classes, $\mathbf{C}'_1$ and $\mathbf{C}'_2$ are calculated using the trials belonging to the set J_s^c (refer equation 3.6).

The composite covariance matrix $\mathbf{C}_L$ is given by:

$$\mathbf{C}_L = \mathbf{C}'_1 + \mathbf{C}'_2 \quad (3.11)$$

On the covariance matrix $\mathbf{C}_L$, an eigen decomposition is performed and is given by:

$$\mathbf{C}_L = \mathbf{H}_c \chi \mathbf{H}_c^T \quad (3.12)$$

where, $\mathbf{H}_c$ denotes the matrix of normalized eigenvectors with corresponding eigenvalues χ . The whitening transformation matrix given by:

$$\mathbf{P} = \sqrt{\chi^{-1}} \mathbf{H}_c^T \quad (3.13)$$

makes the composite covariance matrix $\mathbf{C}_L$ decorrelated,

$$\mathbf{P} \mathbf{C}_L \mathbf{P}^T = \mathbf{P} (\mathbf{C}'_1 + \mathbf{C}'_2) \mathbf{P}^T = \mathbf{I} \quad (3.14)$$

$$\mathbf{C}_1'' = \mathbf{P} \mathbf{C}'_1 \mathbf{P}^T, \quad \mathbf{C}_2'' = \mathbf{P} \mathbf{C}'_2 \mathbf{P}^T \quad (3.15)$$

Now, the common eigenvectors $\mathbf{U}$ for matrices $\mathbf{C}_1''$ and $\mathbf{C}_2''$ are found with corresponding eigenvalues summing to one,

$$\mathbf{C}_1'' = \mathbf{U} \beta_1 \mathbf{U}^T, \quad \mathbf{C}_2'' = \mathbf{U} \beta_2 \mathbf{U}^T, \quad \beta_1 + \beta_2 = \mathbf{I} \quad (3.16)$$

where, $\mathbf{I}$ is the identity matrix. Here, the eigenvector with the largest eigenvalue for $\mathbf{C}_1''$ has the smallest eigenvalue for $\mathbf{C}_2''$ and vice-versa. The projection matrix $\mathbf{V} = \mathbf{U}^T \mathbf{P}$ can be used to map a trial $\mathbf{E}$ and is given as

$$\mathbf{Z} = \mathbf{V} \times \mathbf{E}. \quad (3.17)$$

3.2.1 Feature Extraction

The transformed signals, $\mathbf{Z} \in \Re^{L \times S}$ maximize the variance between the two classes. However, a small number of selected transformed signals, $\mathbf{Z}'$ (first and last 3 rows of $\mathbf{Z}$ [71]) is used for feature extraction. In this chapter, the log-variance of the signal is computed as features and is given as:

$$\mathbf{x}_j = \log \left(\text{var}(\mathbf{Z}') \right) \quad (3.18)$$

where, $\mathbf{x}_j = [x_1, \dots, x_m]^T \in \Re^m$ is a m -dimensional feature vector. Equations (3.17) and (3.18) are repeated for every trial to compute the log-variance features.

The proposed RoCSP algorithm provides robust features by addressing the problem of artifacts in EEG data and use of redundant channels. In the next section, the data sets used in the thesis are described followed by performance evaluation.

3.3 Data Set Description

In this section, the data sets used in the thesis are described in detail.

- **BCI Competition IV Data Set-IIa:**

The BCI competition IV data set-IIa [116] is recorded for nine subjects using twenty-two electrodes sampled at 250 *Hz*. The subjects performed various motor-imagery tasks, namely: left hand (class 1), right hand (class 2), foot (class 3) and tongue (class 4). These four motor-imagery classes have been used to formulate six different binary classification problems. The classification tasks are left vs right hand classification, left hand vs foot classification, left hand vs tongue classification, right hand vs foot classification, right hand vs tongue classification and foot vs tongue motor-imagery classification.

The data is recorded over two sessions on two different days for each subject. Each session is comprised of 6 runs separated by short breaks. One run consists of 48 trials (12 for each of the four possible classes), yielding a total of 288 trials per session. The subjects were sitting in a comfortable armchair in front of a computer screen. At the beginning of a trial ($t = 0$ sec), a fixation cross appeared on the

black screen. In addition, a short acoustic warning tone was presented. After two seconds ($t = 2 \text{ sec}$), a cue in the form of an arrow pointing either to the left, right, down or up (corresponding to one of the four classes left hand, right hand, foot or tongue) appeared and stayed on the screen for 1.25 sec . This prompted the subjects to perform the desired motor-imagery task. The subjects were asked to carry out the motor-imagery task until the fixation cross disappeared from the screen. The data is extracted from a number of selected electrode positions, starting from 0.5 to 2.5 sec after the visual cue. Twenty-two Ag/AgCl electrodes (with interelectrode distances of 3.5 cm) were used to record the EEG signals. The signals were sampled with 250 Hz and bandpass-filtered between 0.5 Hz and 100 Hz .

- **BCI Competition III Data Set-IVa:**

The data set-IVa [18] consists of five subjects ('aa', 'al', 'av', 'aw', 'ay') and is recorded using one hundred eighteen electrodes. The subjects were in a sitting position in front of a computer screen. The data set is recorded over 4 sessions. A visual cue appeared on the computer screen indicating the duration to perform the motor-imageries. The subjects performed left hand, right hand and right foot motor-imagery tasks. In this study, the data for the right hand (class one) and foot (class two) motor-imagery tasks are used for classification. The number of trials in both the training and testing phase varies for each subject. Subject 'aa' has one hundred sixty-eight trials in training and one hundred twelve trials in the testing set, subject 'al' has two hundred twenty four trials in training and fifty-six trials in the testing set, subject 'av' has eighty four trials in the training set and one hundred ninety-six trials in the testing set, subject 'aw' has fifty-six trials in the training and two hundred twenty-four trials in the testing set and subject 'ay' has twenty-eight trials in the training set and two hundred fifty-two trials in the testing set.

3.4 Performance Evaluation

The performance of the proposed RoCSP algorithm is evaluated using two publicly available BCI competition data sets. First, the performance is evaluated on nine subjects with

a moderate number of channels using data set-IIa from BCI competition IV [116] followed by five subjects with a large number of channels using data set-IVa from BCI competition III [117]. The performance of RoCSP is compared with other algorithms that regularize the CSP, including (GLRCSP, CCSP1, CCSP2, DLCSPauto, DLCSPcv, DLCSPcvdiff, SSRCS, TRCSP, WTRCSP and SRCSP). All these algorithms use the Linear Discriminant Analysis (LDA) as a classifier. We show the performance of RoCSP with LDA to highlight its effectiveness over other algorithms including (GLRCSP, CCSP1, CCSP2, DLCSPauto, DLCSPcv, DLCSPcvdiff, SSRCS, TRCSP, WTRCSP and SRCSP) that use the LDA as classifier.

The LDA classifier learns a linear hyper-plane to separate two classes of data [118]. The hyper-plane is learnt with an aim to maximize the interclass distance and minimize the intraclass variance.

3.4.1 Performance of RoCSP - BCI Competition IV Data Set-IIa

In this section, the performance of RoCSP is compared with GLRCSP, CCSP1, CCSP2, DLCSPauto, DLCSPcv, DLCSPcvdiff, SSRCS, TRCSP, WTRCSP and SRCSP for six classification tasks (left vs right hand classification, left vs foot classification, left vs tongue classification, right vs foot classification, right vs tongue classification and foot vs tongue classification). The performance of RoCSP is compared with the baseline CSP algorithm followed by the comparison with the regularized CSP algorithms.

3.4.1.1 Performance Comparison of RoCSP and CSP algorithm

The performance of RoCSP and the baseline CSP algorithm for the six classification tasks is presented in Table 3.1. Based on the subject-wise studies, the experimental results indicate a significant performance improvement. In general, RoCSP is more effective on the subjects with poor performance for the CSP algorithm. For instance, it achieves an improvement of 10% in the case of subject A2 for left-right classification task and an improvement of 15% in the case of subject A4 for left-foot classification task in comparison to the CSP algorithm. On an average, RoCSP achieves better performance for all the six classification tasks. It achieves an increment of 3% for left-foot classification tasks. RoCSP achieves the highest classification performance for left-tongue classification task.

This implies the left and tongue classes are most distinguishable. In all these classification tasks RoCSP uses all the twenty-two channels.

Table 3.1: Classification performance (Subject Mean (SM) in % and Standard Deviation (SD)) obtained for six classification tasks by using CSP-LDA and RoCSP-LDA algorithm for the BCI competition IV data set-IIa.

Sub- ject	Six motor-imagery tasks of the BCI competition IV data set-IIa											
	Left vs Right		Left vs Foot		Left vs Tong.		Right vs Foot		Right vs Tong.		Foot vs Tong.	
	CSP	RoCSP	CSP	RoCSP	CSP	RoCSP	CSP	RoCSP	CSP	RoCSP	CSP	RoCSP
A1	88.89	88.89	97.92	99.31	96.53	97.92	99.31	99.31	100	100	72.22	72.92
A2	51.39	61.11	69.44	70.14	67.36	71.53	79.86	79.86	63.89	66.67	70.83	70.83
A3	96.53	96.53	93.06	93.06	94.44	95.14	91.67	93.06	96.53	96.53	68.75	70.83
A4	70.14	70.14	75.69	90.97	90.28	90.28	88.19	89.58	84.72	84.72	55.55	66.67
A5	54.86	61.81	61.11	69.44	69.44	73.61	64.58	64.58	65.97	70.83	67.36	68.75
A6	71.53	71.53	56.94	56.94	71.53	75.00	63.19	63.19	68.75	68.75	66.67	67.36
A7	81.25	81.25	98.61	98.61	97.22	97.22	99.31	99.31	97.22	97.92	81.94	81.94
A8	93.75	95.83	86.81	87.50	90.28	95.83	88.88	89.58	90.97	90.97	80.56	85.42
A9	93.75	93.75	95.14	95.83	95.14	95.14	81.94	89.58	86.11	86.11	90.97	91.67
SM	78.01	80.09	81.64	84.64	85.80	87.96	84.10	85.34	83.80	84.72	72.76	75.15
SD	17.01	14.36	16.23	15.26	12.54	11.18	13.23	13.49	14.16	13.05	10.36	8.95

3.4.1.2 Performance Comparison for Left-Right Classification Task

In this section, the performance of the proposed RoCSP algorithm is compared with GLRCSP, CCSP1, CCSP2, DLCSPauto, DLCSPcv, DLCSPcvdiff, SSRCSPP, TRCSP, WTRCSP and SRCSP for left-right classification task. Table 3.2 shows the classification accuracies of the nine subjects for all the specified algorithms. It can be seen from the table that on an average RoCSP performs better than all the algorithms for left-right classification task. The next best performing algorithm in this case is TRCSP. Compared to the algorithms (DLCSPauto, DLCSPcv and DLCSPcvdiff) which regularize the class covariance matrix towards identity matrix, RoCSP achieves an improvement of 2-3%. Subject-wise study shows that diagonal loading is not efficient in case of left right classification task. Compared to algorithms (CCSP1, CCSP2, SSRCSPP, GLRCSP) that use data from other subjects to construct the covariance matrix, RoCSP achieves an improvement of 3-4%. It can be observed from the Table 3.2 that even after using data from other subjects CCSP2 and SSRCSPP achieve low performance in the case of subject A2. CCSP1 and GLRCSP are able to achieve better performance in case of subject A2 but perform poorly in the case of subject A5 and A6. In all these cases, RoCSP achieves better performance without considering data from other subjects. Compared to

algorithms (TRCSP and WTRCSP) which use Tikhonov regularization, RoCSP achieves an improvement of 2% on an average. Finally, compared to SRCSP which considers the location of electrodes, RoCSP achieves better subject-wise performance.

Table 3.2: Classification performance (SM in % and SD) obtained for each subject for the standard CSP and its variants and RoCSP-LDA on the intersession(different days) for left vs right hand task -BCI competition IV data set-IIa.

Subjects	A1	A2	A3	A4	A5	A6	A7	A8	A9	SM $\pm$ SD
CSP	88.89	51.39	96.53	70.14	54.86	71.53	81.25	93.75	93.75	78.01 $\pm$ 17.01
GLRCSP	86.11	58.33	93.75	67.36	55.56	65.28	81.25	93.75	88.19	76.62 $\pm$ 15.10
CCSP1	86.11	60.42	93.75	56.94	49.31	65.28	81.25	93.75	88.19	75.00 $\pm$ 17.08
CCSP2	88.89	53.47	97.22	70.14	54.17	68.06	79.17	95.14	90.28	77.39 $\pm$ 16.81
DLCSPauto	88.89	51.39	96.53	70.14	56.94	71.53	81.94	93.75	93.75	78.32 $\pm$ 16.69
DLCSPcv	88.89	50.69	96.53	70.14	55.56	62.50	81.25	93.75	86.81	76.24 $\pm$ 17.04
DLCSPcvdiff	88.89	50.69	96.53	70.14	55.56	62.50	81.25	93.75	86.81	76.24 $\pm$ 17.04
SSRCSP	88.89	53.47	97.22	70.14	56.25	68.75	79.17	97.22	90.28	77.93 $\pm$ 16.70
TRCSP	88.89	54.17	96.53	70.83	62.50	67.36	81.25	95.83	91.67	78.78 $\pm$ 15.59
WTRCSP	88.89	54.86	96.53	70.14	65.97	61.81	81.25	95.83	90.97	78.47 $\pm$ 15.66
SRCSP	88.89	52.78	96.53	70.14	56.25	65.97	81.94	95.14	91.67	77.70 $\pm$ 16.86
RoCSP-LDA	88.89	61.11	96.53	70.14	61.81	71.53	81.25	95.83	93.75	80.09 $\pm$ 14.36

3.4.1.3 Performance Comparison for Left-Foot Classification Task

In this section, the performance of the proposed RoCSP algorithm is compared with GLRCSP, CCSP1, CCSP2, DLCSPauto, DLCSPcv, DLCSPcvdiff, SSRCSP, TRCSP, WTRCSP and SRCSP for left-foot classification task. Table 3.3 shows the classification accuracies of the nine subjects for all the specified algorithms. It can be observed from the table that in the case of left-foot classification task, RoCSP algorithm achieves better average performance than the compared algorithms. The next best performing algorithm is SRCSP closely followed by WTRCSP. Compared to algorithms (CCSP1, CCSP2, SSRCSP, GLRCSP), which use the data of other subjects to construct the covariance matrix, RoCSP achieves an improvement of 2-3%. In comparison to algorithms that use diagonal loading to construct the class covariance matrix, RoCSP achieves an improvement of 3-4%. Compared to algorithms that employ Tikhonov regularization, RoCSP achieves an improvement of 2% on an average.

Table 3.3: Classification performance (SM in % and SD) obtained for each subject for the standard CSP and its variants and RoCSP-LDA on the intersession(different days) for left vs foot task- BCI competition IV data set-IIa.

Subjects	A1	A2	A3	A4	A5	A6	A7	A8	A9	SM $\pm$ SD
CSP	97.92	69.44	93.06	75.69	61.11	56.94	98.61	86.81	95.14	81.64 $\pm$ 16.23
GLRCSP	93.75	69.44	93.06	75.69	61.81	67.36	98.61	86.81	93.75	82.25 $\pm$ 13.77
CCSP1	96.53	69.44	93.06	75.69	63.89	56.94	98.61	86.81	93.75	81.64 $\pm$ 15.50
CCSP2	96.53	72.22	93.06	71.53	68.06	53.47	98.61	84.03	93.75	81.25 $\pm$ 15.65
DLCSPauto	97.92	70.83	93.06	75.69	61.11	56.94	98.61	86.81	94.44	81.71 $\pm$ 16.04
DLCSPcv	93.75	69.44	93.06	75.69	53.47	56.94	98.61	86.81	95.14	80.32 $\pm$ 17.12
DLCSPcvdiff	96.53	72.92	93.06	75.69	61.11	56.94	93.06	86.81	94.44	81.17 $\pm$ 15.06
SSRCSP	97.22	70.83	93.06	74.31	69.44	63.89	98.61	86.81	94.44	83.18 $\pm$ 13.53
TRCSP	95.14	69.44	92.36	71.53	54.17	56.94	98.61	87.50	95.14	80.09 $\pm$ 17.31
WTRCSP	96.53	69.44	93.06	86.11	66.67	56.94	98.61	86.81	95.14	83.26 $\pm$ 15.12
SRCSP	94.44	72.22	93.75	86.11	63.19	63.89	98.61	86.81	95.14	83.80 $\pm$ 13.83
RoCSP-LDA	99.31	70.14	93.06	90.97	69.44	56.94	98.61	87.50	95.83	84.64 $\pm$ 15.26

3.4.1.4 Performance Comparison for Left-Tongue Classification Task

In this section, the performance comparison of the proposed RoCSP algorithm with GLRCSP, CCSP1, CCSP2, DLCSPauto, DLCSPcv, DLCSPcvdiff, SSRCSP, TRCSP, WTRCSP and SRCSP for left-tongue classification task is presented. Table 3.4 shows the classification accuracies of the nine subjects for all the specified algorithms. It can be observed from the Table 3.4 that on an average RoCSP performs better than all the compared algorithms. The next best performing algorithm is CCSP2 which uses data from other subjects. It performs better than RoCSP by 1% in the case of subject A1 and A9, however, RoCSP performs significantly better in the case of subjects A2, A6 and A8. Compared to other algorithms (CCSP1, SSRCSP, GLRCSP) that use data from other subjects, RoCSP achieves an increment of 2% on an average. Compared to algorithms (DLCSPauto, DLCSPcv, DLCSPcvdiff) which employ diagonal loading, RoCSP achieves an increment of 2% in case of DLCSPauto and DLCSPcv and 6% in case of DLCSPcvdiff. Compared to algorithms (TRCSP, WTRCSP) that employ Tikhonov regularization, RoCSP achieves an improvement of 2% on an average.

3.4.1.5 Performance Comparison for Right-Foot Classification Task

In this section, the performance comparison of the proposed RoCSP algorithm with GLRCSP, CCSP1, CCSP2, DLCSPauto, DLCSPcv, DLCSPcvdiff, SSRCSP, TRCSP,

Table 3.4: Classification performance (SM in % and SD) obtained for each subject for the standard CSP and its variants and RoCSP-LDA on the intersession(different days) for left vs tongue task- BCI competition IV data set-IIa.

Subjects	A1	A2	A3	A4	A5	A6	A7	A8	A9	SM $\pm$ SD
CSP	96.53	67.36	94.44	90.28	69.44	71.53	97.22	90.28	95.14	85.80 $\pm$ 12.54
GLRCSP	97.22	61.81	94.44	90.28	65.28	71.53	97.22	90.28	96.53	84.95 $\pm$ 14.51
CCSP1	97.22	58.33	94.44	90.28	69.44	71.53	97.22	90.28	96.53	85.03 $\pm$ 14.63
CCSP2	98.61	65.28	94.44	88.89	73.61	71.53	97.22	90.28	96.53	86.27 $\pm$ 12.68
DLCSPauto	96.53	66.67	94.44	90.97	69.44	71.53	97.22	90.97	95.83	85.96 $\pm$ 12.80
DLCSPcv	96.53	62.50	94.44	90.28	70.14	71.53	97.22	90.28	95.14	85.34 $\pm$ 13.40
DLCSPcvdiff	95.83	59.72	85.42	80.56	56.94	63.89	95.14	90.28	95.14	80.32 $\pm$ 15.99
SSRCSP	97.92	66.67	93.75	87.50	70.83	71.53	97.22	90.28	95.14	85.65 $\pm$ 12.47
TRCSP	96.53	60.42	94.44	90.28	72.22	71.53	97.22	90.28	95.14	85.34 $\pm$ 13.59
WTRCSP	96.53	62.50	94.44	90.28	70.14	72.22	97.22	90.28	95.14	85.42 $\pm$ 13.32
SRCSP	98.61	61.81	94.44	88.89	61.81	68.06	97.22	90.28	95.14	84.03 $\pm$ 15.51
RoCSP-LDA	97.92	71.53	95.14	90.28	73.61	75.00	97.22	95.83	95.14	87.96 $\pm$ 11.18

WTRCSP and SRCSP for right-foot classification task is presented. Table 3.5 shows the classification accuracies of the nine subjects for all the specified algorithms. It can be seen from the table that RoCSP performs better than the compared algorithms on an average. The next best performing algorithm in case of right-foot task is SRCSP. Here, compared to algorithms (DLCSPauto, DLCSPcv, DLCSPcvdiff) that employ diagonal loading, RoCSP achieves an average improvement of 1-2% compared to DLCSPauto and DLCSPcv and 6% compared to DLCSPcvdiff. In comparison to algorithms (CCSP1, CCSP2, SSRCSP, GLRCSP) which use data from other subjects, RoCSP achieves an improvement of 2-3%. Compared to algorithms (TRCSP, WTRCSP) that employ Tikhonov regularization, RoCSP achieves an improvement of 1.5% on an average.

3.4.1.6 Performance Comparison for Right-Tongue Classification Task

In this section, the performance comparison of the proposed RoCSP algorithm with GLRCSP, CCSP1, CCSP2, DLCSPauto, DLCSPcv, DLCSPcvdiff, SSRCSP, TRCSP, WTRCSP and SRCSP for right-tongue classification task is presented. Table 3.6 shows the classification accuracies of the nine subjects for all the specified algorithms. The best performing algorithm in this case is TRCSP. The proposed RoCSP algorithm performs nearly as well as TRCSP in this case. It can also be observed from the table that RoCSP performs better than the algorithms that employ diagonal loading as well as algorithms that use data from other subjects for regularization.

Table 3.5: Classification performance (SM in % and SD) obtained for each subject for the standard CSP and its variants and RoCSP-LDA on the intersession(different days) for right vs foot task- BCI competition IV data set-IIa.

Subjects	A1	A2	A3	A4	A5	A6	A7	A8	A9	SM $\pm$ SD
CSP	99.31	79.86	91.67	88.19	64.58	63.19	99.31	88.89	81.94	84.10 $\pm$ 13.23
GLRCSP	97.22	68.75	91.67	88.19	61.81	63.19	99.31	88.89	86.11	82.79 $\pm$ 14.40
CCSP1	97.22	79.86	91.67	88.19	55.56	63.19	99.31	88.89	86.11	83.33 $\pm$ 14.87
CCSP2	98.61	75.00	91.67	88.19	59.72	63.19	96.53	88.89	84.03	82.87 $\pm$ 13.98
DLCSPauto	99.31	79.86	91.67	88.19	64.58	63.89	99.31	88.89	82.64	84.26 $\pm$ 13.08
DLCSPcv	97.22	79.86	91.67	88.19	63.19	63.19	99.31	88.89	81.94	83.72 $\pm$ 13.21
DLCSPcvdiff	97.92	54.17	91.67	88.19	54.86	63.19	90.28	88.89	84.03	79.24 $\pm$ 16.96
SSRCSP	100.00	76.39	91.67	88.19	57.64	64.58	95.83	88.89	86.11	83.26 $\pm$ 14.25
TRCSP	99.31	78.47	91.67	84.03	65.97	63.89	95.14	88.89	81.94	83.26 $\pm$ 12.25
WTRCSP	98.61	80.56	91.67	88.19	63.89	65.28	95.14	88.89	87.50	84.41 $\pm$ 12.33
SRCS	98.61	80.56	91.67	84.03	67.36	65.28	95.14	88.89	88.19	84.41 $\pm$ 11.59
RoCSP-LDA	99.31	79.86	93.06	89.58	64.58	63.19	99.31	89.58	89.58	85.34 $\pm$ 13.49

Table 3.6: Classification performance (SM in % and SD) obtained for each subject for the standard CSP and its variants and RoCSP-LDA on the intersession(different days) for right vs tongue task- BCI competition IV data set-IIa.

Subjects	A1	A2	A3	A4	A5	A6	A7	A8	A9	SM $\pm$ SD
CSP	100.00	63.89	96.53	84.72	65.97	68.75	97.22	90.97	86.11	83.80 $\pm$ 14.16
GLRCSP	100.00	63.89	97.92	84.72	61.81	66.67	97.92	90.97	86.11	83.33 $\pm$ 15.37
CCSP1	100.00	63.89	97.92	84.72	62.50	65.28	97.92	90.97	86.11	83.26 $\pm$ 15.45
CCSP2	100.00	69.44	97.22	84.72	63.19	65.97	96.53	90.97	86.11	83.80 $\pm$ 14.19
DLCSPauto	100.00	65.28	96.53	85.42	66.67	69.44	97.22	90.97	86.81	84.26 $\pm$ 13.73
DLCSPcv	100.00	63.89	96.53	84.72	61.11	63.19	97.22	90.97	86.11	82.64 $\pm$ 15.76
DLCSPcvdiff	100.00	63.89	98.61	84.72	61.81	63.89	97.22	90.97	86.11	83.02 $\pm$ 15.78
SSRCSP	100.00	63.89	98.61	84.72	61.81	66.67	96.53	90.97	86.11	83.26 $\pm$ 15.30
TRCSP	100.00	68.75	100.00	86.11	71.53	65.28	97.22	90.97	85.42	85.03 $\pm$ 13.56
WTRCSP	100.00	69.44	99.31	85.42	65.28	68.75	95.83	90.97	85.42	84.49 $\pm$ 13.58
SRCS	100.00	66.67	97.92	85.42	60.42	65.28	96.53	90.97	86.81	83.33 $\pm$ 15.28
RoCSP-LDA	100.00	66.67	96.53	84.72	70.83	68.75	97.92	90.97	86.11	84.72 $\pm$ 13.05

3.4.1.7 Performance Comparison for Foot-Tongue Classification Task

In this section, the performance of the proposed RoCSP algorithm is compared with GLRCSP, CCSP1, CCSP2, DLCSPauto, DLCSPcv, DLCSPcvdiff, SSRCS, TRCSP, WTRCSP and SRCSP for foot-tongue classification task. Table 3.7 shows the classification accuracies of the nine subjects for all the specified algorithms. It can be observed from the table that RoCSP performs better than the compared algorithms. The next best performing algorithm in this case is SRCSP. Compared to algorithms (CCSP1, CCSP2, SSRCS, GLRCSP) that use data from other subjects to construct a generic covariance matrix, RoCSP achieves an improvement in performance by 2-3%. In comparison to algorithms (DLCSPauto, DLCSPcv, DLCSPcvdiff) which employ diagonal leading, RoCSP achieves a performance increment of 2-3% on an average.

Table 3.7: Classification performance (SM in % and SD) obtained for each subject for the standard CSP and its variants and RoCSP-LDA on the intersession(different days) for foot vs tongue task- BCI competition IV data set-IIa.

Subjects	A1	A2	A3	A4	A5	A6	A7	A8	A9	SM $\pm$ SD
CSP	72.22	70.83	68.75	55.56	67.36	66.67	81.94	80.56	90.97	72.76 $\pm$ 10.36
GLRCSP	72.22	71.53	68.75	55.56	67.36	68.06	81.94	86.11	88.89	73.38 $\pm$ 10.51
CCSP1	72.22	71.53	68.75	55.56	56.94	68.06	81.94	86.11	90.28	72.38 $\pm$ 12.01
CCSP2	72.92	72.22	68.75	59.03	59.72	65.97	81.94	82.64	88.89	72.45 $\pm$ 10.38
DLCSPauto	71.53	70.83	68.75	56.25	68.75	67.36	81.94	81.25	91.67	73.15 $\pm$ 10.32
DLCSPcv	72.22	61.81	68.75	55.56	67.36	66.67	81.94	80.56	89.58	71.60 $\pm$ 10.69
DLCSPcvdiff	72.22	67.36	68.75	55.56	67.36	66.67	81.94	79.17	89.58	72.07 $\pm$ 10.07
SSRCSP	74.31	75.69	68.75	55.56	69.44	69.44	81.94	79.17	86.81	73.46 $\pm$ 9.11
TRCSP	72.22	68.06	68.75	56.25	67.36	68.06	81.94	84.72	88.89	72.92 $\pm$ 10.31
WTRCSP	72.22	72.92	68.75	59.03	70.14	67.36	81.94	81.25	88.89	73.61 $\pm$ 9.02
SRCSP	72.22	72.92	68.75	59.72	68.75	67.36	81.94	85.42	88.19	73.92 $\pm$ 9.37
RoCSP-LDA	72.92	70.83	70.83	66.67	68.75	67.36	81.94	85.42	91.67	75.15 $\pm$ 8.95

The summarized results for all the classification tasks for all the algorithms is presented in Table 3.8. It can be seen from the table that RoCSP performs better than all the algorithms in all the classification tasks except the right-tongue classification task. In the case of this task, the performance of RoCSP is close to the best performing TRCSP algorithm.

Table 3.8: Comparison of classification performance (SM in % and SD) obtained for six classification tasks for CSP algorithm and its variants with the proposed RoCSP-LDA on the intersession (different days) data set II-a from BCI competition-IV

(a) Left vs right		
Algorithms	SM $\pm$ SD	
CSP	78.01 $\pm$ 17.01	
GLRCSP	76.62 $\pm$ 15.10	
CCSP1	75.00 $\pm$ 17.08	
CCSP2	77.39 $\pm$ 16.81	
DLCSPPauto	78.32 $\pm$ 16.69	
DLCSPPcv	76.24 $\pm$ 17.04	
DLCSPPdiff	76.24 $\pm$ 17.04	
SSRCSP	77.93 $\pm$ 16.70	
TRCSP	78.78 $\pm$ 15.59	
WTRCSP	78.47 $\pm$ 15.66	
SRCSP	77.70 $\pm$ 16.86	
RoCSP-LDA	80.09 $\pm$ 14.36	

(b) Left vs foot		
Algorithms	SM $\pm$ SD	
CSP	81.64 $\pm$ 16.23	
GLRCSP	82.25 $\pm$ 13.77	
CCSP1	81.64 $\pm$ 15.50	
CCSP2	81.25 $\pm$ 15.65	
DLCSPPauto	81.71 $\pm$ 16.04	
DLCSPPcv	80.32 $\pm$ 17.12	
DLCSPPdiff	81.17 $\pm$ 15.06	
SSRCSP	83.18 $\pm$ 13.53	
TRCSP	80.09 $\pm$ 17.31	
WTRCSP	83.26 $\pm$ 15.12	
SRCSP	83.80 $\pm$ 13.83	
RoCSP-LDA	84.64 $\pm$ 15.26	

(c) Left vs tongue		
Algorithms	SM $\pm$ SD	
CSP	85.80 $\pm$ 12.54	
GLRCSP	84.95 $\pm$ 14.51	
CCSP1	85.03 $\pm$ 14.63	
CCSP2	86.27 $\pm$ 12.68	
DLCSPPauto	85.96 $\pm$ 12.80	
DLCSPPcv	85.34 $\pm$ 13.40	
DLCSPPdiff	80.32 $\pm$ 15.99	
SSRCSP	85.65 $\pm$ 12.47	
TRCSP	85.34 $\pm$ 13.59	
WTRCSP	85.42 $\pm$ 13.32	
SRCSP	84.03 $\pm$ 15.51	
RoCSP-LDA	87.96 $\pm$ 11.18	

(d) Right vs foot		
Algorithms	SM $\pm$ SD	
CSP	84.10 $\pm$ 13.23	
GLRCSP	82.79 $\pm$ 14.40	
CCSP1	83.33 $\pm$ 14.87	
CCSP2	82.87 $\pm$ 13.98	
DLCSPPauto	84.26 $\pm$ 13.08	
DLCSPPcv	83.72 $\pm$ 13.21	
DLCSPPdiff	79.24 $\pm$ 16.96	
SSRCSP	83.26 $\pm$ 14.25	
TRCSP	83.26 $\pm$ 12.25	
WTRCSP	84.41 $\pm$ 12.33	
SRCSP	84.41 $\pm$ 11.59	
RoCSP-LDA	85.34 $\pm$ 13.49	

(e) Right vs tongue		
Algorithms	SM $\pm$ SD	
CSP	83.80 $\pm$ 14.16	
GLRCSP	83.33 $\pm$ 15.37	
CCSP1	83.26 $\pm$ 15.45	
CCSP2	83.80 $\pm$ 14.19	
DLCSPPauto	84.26 $\pm$ 13.73	
DLCSPPcv	82.64 $\pm$ 15.76	
DLCSPPdiff	83.02 $\pm$ 15.78	
SSRCSP	83.26 $\pm$ 15.30	
TRCSP	85.03 $\pm$ 13.56	
WTRCSP	84.49 $\pm$ 13.58	
SRCSP	83.33 $\pm$ 15.28	
RoCSP-LDA	84.72 $\pm$ 13.05	

(f) Foot vs tongue		
Algorithms	SM $\pm$ SD	
CSP	72.76 $\pm$ 10.36	
GLRCSP	73.38 $\pm$ 10.51	
CCSP1	72.38 $\pm$ 12.01	
CCSP2	72.45 $\pm$ 10.38	
DLCSPPauto	73.15 $\pm$ 10.32	
DLCSPPcv	71.60 $\pm$ 10.69	
DLCSPPdiff	72.07 $\pm$ 10.07	
SSRCSP	73.46 $\pm$ 9.11	
TRCSP	72.92 $\pm$ 10.31	
WTRCSP	73.61 $\pm$ 9.02	
SRCSP	73.92 $\pm$ 9.37	
RoCSP-LDA	75.15 $\pm$ 8.95	

3.4.2 Performance of RoCSP - BCI Competition III Data Set-IVa

In this section, the performance of RoCSP is evaluated using BCI competition III data set-IVa. The data set-IVa [18] consists of five subjects ('aa', 'al', 'av', 'aw', 'ay') and is recorded using one hundred eighteen electrodes. The performance of RoCSP for the five subjects ('aa', 'al', 'av', 'aw', 'ay') is presented in Table 3.9. It shows the classification performance of CSP using all the channels, classification performance of RoCSP using the selected channels and number of selected channels. It can be observed from the Table 3.9 that the RoCSP algorithm provides an increment of 16% over the CSP algorithm. It should also be noted that on an average, it uses 17 number of channels which is significantly lesser than 118 channels and achieves a significant performance improvement. Subject 'ay' shows the maximum increment of 40% over the CSP algorithm using 12 channels in comparison to 118 channels of CSP. In case of Subject 'aa', a total of 73 trials were eliminated while computing the RoCSP projection matrix. The value of trial selection threshold is 0.8 and the frequency parameter is 15. However, these trials were only eliminated in the process of channel selection while computing the projection matrix. They are used as a part of training dataset. Further, a total of 97 redundant channels were eliminated from 118 channels. The value of channel selection threshold is 0.51.

Table 3.9: Classification performance (Subject Mean (SM) in % and Standard Deviation(SD)) with selected channels using RoCSP, classification performance with all channels using CSP, number of selected channels for each subject, threshold δ for the BCI competition III data set-IVa.

Subjects	No. of Selected Channels using RoCSP	Threshold δ	Testing Accuracy	
			CSP	RoCSP
aa	21	0.51	66.07	71.43
al	21	0.49	96.43	100
av	17	0.36	47.45	69.90
aw	14	0.35	71.88	81.69
ay	12	1.21	49.6	90.47
SM			66.28	82.69
SD			19.83	12.76

The performance of RoCSP is compared with GLRCSP, CCSP1, CCSP2, DLCSPauto, DLCSPcv, DLCSPcvdiff, SSRCSPP, TRCSP, WTRCSP and SRCSP in Table 3.10. All

these algorithms use data from 118 channels. It can be seen from the table that on an average RoCSP performs better than the compared algorithm using lesser number of channels. The next best performing algorithm in this case is SRCSP. Compared to algorithms (CCSP1, CCSP2, SSRCS, GLRCSP) that use data from other subjects to construct a generic covariance matrix, RoCSP achieves an improvement in performance by 5-6% on an average. In comparison to algorithms (DLCSPauto, DLCSPcv, DLCSPcvdiff) which employ diagonal leading, RoCSP achieves a performance increment of 11% on an average. Compared to algorithms (TRCSP, WTRCSP) that use Tikhonov regularization, RoCSP achieves an improvement of 6% on an average.

Table 3.10: Classification performance (SM in % and SD) obtained for each subject for the standard CSP and its variants and RoCSP-LDA on BCI competition III data set-IVa.

Algorithm	Subjects					SM $\pm$ SD
	aa	al	av	aw	ay	
CSP	66.07	96.43	47.45	71.88	49.6	66.28 $\pm$ 19.83
GLRCSP	72.32	96.43	66.84	67.86	89.29	78.54 $\pm$ 13.46
CCSP1	66.96	96.43	63.27	71.88	84.92	76.69 $\pm$ 13.74
CCSP2	65.18	96.43	45.41	71.88	49.6	65.70 $\pm$ 20.33
DLCSPauto	66.96	96.43	46.94	71.43	50	66.35 $\pm$ 19.84
DLCSPcv	64.29	96.43	52.04	71.88	82.54	73.43 $\pm$ 16.99
DLCSPcvdiff	69.64	98.21	55.10	71.88	82.54	75.47 $\pm$ 16.03
SSRCSP	70.54	96.43	53.57	71.88	75.39	73.56 $\pm$ 15.31
TRCSP	71.43	96.43	63.27	71.88	86.9	77.98 $\pm$ 13.38
WTRCSP	69.64	98.21	54.59	71.88	85.32	75.92 $\pm$ 16.55
SRCSP	72.32	96.43	60.2	77.68	86.51	78.62 $\pm$ 13.77
RoCSP-LDA	71.43	100	69.90	81.69	90.47	82.69 $\pm$ 12.76

The proposed RoCSP feature extraction algorithm does not have the ability to handle non-stationarity in EEG data. In the next section, the effect of non-stationarity in EEG data on the proposed RoCSP algorithm is illustrated.

3.4.3 Illustration of Effect of Non-stationarity in EEG Data on RoCSP

In this section, the effect of non-stationarity in EEG signal on RoCSP is illustrated using two (subject A2 and subject A8) from BCI competition (IV-IIa, left-right classification

task) [116]. In the data set IV-IIa, two sessions on different days were recorded for each subject, where, the first session data is used for training and the second session data is used for testing. In the BCI competition data set IV-IIa, the subject A8 can be classified easily using a simple LDA classifier using the log-variance features extracted from RoCSP and this clearly indicates that the signal is stationary. For the subject A2, the performance of LDA classifier is only 61.11% and this may be due to the presence of non-stationarity in the data. In this case, log-variance features extracted using RoCSP do not provide discriminating features and hence the LDA fails to classify accurately.

Stationary EEG case: For the subject A8 from the BCI competition data set IV-IIa, the features are extracted using RoCSP and then the LDA classifier is used to classify the EEG signals. The data from the first day is used to find the RoCSP projection matrix and the same projection matrix is used on the data of the second day to obtain the spatially filtered signals. Finally, the log-variances of the three most dominant pair of spatially filtered signals, obtained by RoCSP, are used as features. These feature pair are most suitable in discriminating the two classes of motor-imagery tasks. The scatter plots of the first (x_1 and x_6), second (x_2 and x_5), and third (x_3 and x_4), most dominant pair of features for the training data (day1 signals) as well as testing data (day2 signals) are provided in Fig. 3.4. The scatter plot using the most discriminate pairs (x_1 and x_6) is shown in Fig. 3.4-a. From the figure, one can observe that there is a clear separability between the trials of the two classes. The same pair of features when used for the testing data also provide better separability as shown in Fig. 3.4-b. This clearly indicates that the subject A8 does not have any effective non-stationarity in the EEG signal. However, other feature pairs like (x_2 and x_5) are also needed, as there may be samples (Sample 1) as pointed in Figs. 3.4-b and 3.4-d which may be separable in second most or third most dominant feature space. Due to clear separability, for this data set (subject A8) one can get good classification performance using RoCSP-LDA.

Non-Stationary EEG Case: Next, the subject A2 from the BCI competition data set IV-IIa is considered. Similar to the subject A8, the data from the first day is used to find the RoCSP projection matrix and the same projection matrix is used on the data of the second day to obtain the spatially filtered signals. Finally, the log-variances of the three most dominant pair of spatially filtered signals, obtained by RoCSP, are used as

Figure 3.4: Scatter plot for train and test data using three most dominant pair of features obtained by RoCSP for subject A8 of left-right classification, BCI competition IV data set-IIa. The plus sign and circle denote the features of the left hand and right hand movement, respectively. The dashed line represents the linear hyperplane obtained by the train data.

features. The scatter plots of the first (x_1 and x_6), second (x_2 and x_5), and third (x_3 and x_4), most dominant pair of features for the training data (day1 signals) as well as testing data (day2 signals) are provided in Fig. 3.5. Unlike the subject A8, the scatter plot of

Figure 3.5: Scatter plot for train and test data using three most dominant pair of features obtained by RoCSP for subject A2 of left-right classification, BCI competition IV data set-IIa. The plus sign and circle denote the features of the left hand and right hand movement, respectively. The dashed line represents the linear hyperplane obtained by the train data.

the most discriminate features in subject A2 (shown in Fig. 3.5-a) are not separable in the training data. The separability becomes even poorer when plotted for the testing data as shown in Fig. 3.5-b. Similarly, the rest of the dominant pairs shown in Fig.

3.5-(c-f) do not show any separability. For the subject A2, RoCSP is not able to provide discriminant features. This may be due to the change in the mental state of the subject during the experiment or the presence of non-stationarity in the EEG signal.

3.5 Summary

In this chapter, a Robust Common Spatial Pattern (RoCSP) feature extraction algorithm is proposed that addresses the problem of presence of artifacts in EEG data and use of redundant channels while recording data. Z -score has been used for removal of trials and Cohen's d has been employed for discarding channels that do not carry distinguishable information to discriminate any two classification tasks. The performance of RoCSP algorithm has been evaluated using two publicly available BCI competition data sets (Data sets IVa and IIa from BCI competition III and IV respectively). It gives best accuracy for left-tongue classification task. The results have been compared with algorithms that regularize the CSP algorithm. Subject-wise results show that RoCSP performs better than the compared algorithms. The reduction in the number of channels is significant in the case of data set IVa from BCI competition III which has a large number of channels.

The performance of CSP based algorithms is significantly influenced by the presence of non-stationarity in EEG data. Hence, there is a need to propose an algorithm that has the ability to handle non-stationarity in EEG signals. In the next chapter, a classifier is developed that has the capability to handle non-stationarity.

Chapter 4

A Self-Regulated Interval Type-2 Neuro-Fuzzy Inference System

In the previous chapter, a Robust Common Spatial Pattern (RoCSP) algorithm has been proposed that provides robust features for classification. However, it cannot handle the inherent non-stationarity in EEG signals. Non-stationary nature of the EEG signals arises due to the trial-to-trial variation for a task within a session and also variations that arise across different sessions. These variations are due to the delays in responses to the visual cues, change of visual tasks, change in placement of electrodes, etc. [23]. These variations deteriorate the performance of most of the models that inherently assume that the data is stationary [24–26].

Figure 4.1: Brain signal processing framework.

This chapter focuses on the development of a classifier that is capable of handling non-stationarity (highlighted in Fig. 4.1). Existing classification algorithms like artificial neural networks, type-1 fuzzy inference systems, etc. do not have the capability to handle

uncertain non-stationary data [119, 120]. Recently, it has been shown that type-2 fuzzy sets proposed by Zadeh [121] can handle non-stationarity when cast in an uncertainty framework [122]. The membership function of a type-2 fuzzy set is fuzzy and has an additional third dimension compared to the traditional type-1 fuzzy set. This helps type-2 fuzzy set to capture the non-stationarity in the data as an uncertainty. However, its usage is computationally expensive due to the need for a type-reduction operation. In order to remove this computational burden, type-2 fuzzy sets have been simplified to interval type-2 fuzzy sets [123] where uncertainty is represented as a bounded interval. Based on these fuzzy sets, a class of algorithms known as Interval Type-2 Neuro-Fuzzy Inference Systems (IT2NFIS) has been developed [119, 124–136].

In the early works of IT2NFIS, a fixed architecture has been proposed [124–130]. However, this is not suitable for time-varying data like EEG signals [131–133]. To handle the time varying data, IT2NFIS have been proposed that have the capability to evolve both their structure and parameters based on the data. In [119] the structure and parameters have been evolved in two separate phases and in [137–143], the structure and parameters are evolved simultaneously. Early works on handling non-stationarity using interval type-2 fuzzy systems are presented in [144–147]. In [144], Herman et al., showed that interval type-2 classifiers can be used to handle the non-stationary and uncertain EEG data in BCI. It uses the short term Fourier transform for feature extraction and the employed fuzzy rules are of Mamdani type. Further, in [145, 146] the approach has been extended for an online learning approach with power spectral density based feature extraction. Here, type-2 classifier was designed using a two-phased approach. In the first phase, an initial rule base was designed and in the second phase the parameters were learnt using a gradient descent algorithm. In [147], an approach to classify motor imagery EEG signals using an interval type-2 fuzzy logic system with wavelet transform is presented. Here, the presence of artifacts in training influences the performance significantly. However, these algorithms lack the ability to self-regulate their own knowledge, which might result in over-training. Meta-cognition, which facilitates self-regulation [148] has been shown to be effective in evolving the architecture efficiently [149–155].

In this chapter, a Self-Regulated Interval Type-2 Neuro-Fuzzy Inference System (SRIT-2NFIS) is developed to handle the non-stationarity. It handles the non-stationarity by

modeling it as uncertainty in the classifier. A five-layered modified Takagi-Sugeno-Kang interval type-2 fuzzy inference mechanism forms the structure and the learning algorithm is a self-regulatory learning mechanism. Here, uncertainty is modeled as an additive uncertainty using Gaussian membership function with uncertain mean and fixed standard deviation. The learning algorithm proposed in this chapter is a self-regulating learning algorithm. The learning starts with zero rules and as a sample is presented to the network, the learning algorithm monitors the knowledge in the sample and decides whether to delete it, learn it or keep it for future use. The original inspiration of the self-regulating learning algorithm is Nelson and Narens model of meta-cognition [1]. It uses hinge loss error, spherical potential and maximum rule contribution to monitor the knowledge present in the sample. When a new rule is added to the network, a regularized projection based learning is employed to compute the optimal weights of the network such that it avoids over-fitting. This work has been published in [156].

The main advantage of SRIT2NFIS over other algorithms [137–143] proposed in the literature is its self-regulating nature. SRIT2NFIS has the ability to regulate its knowledge by deciding which samples should be learnt and which samples should be discarded to avoid overfitting.

The performance of SRIT2NFIS is evaluated on a set of benchmark binary/multi-category classification problems with balanced/imbalanced data sets. The performance for classification problems is compared with Evolving Fuzzy Rule based Classifier (eClass) [157], On-line Sequential Fuzzy Extreme Learning Machine (OS-F-ELM) [158], Sequential Adaptive Fuzzy Inference System (SAFIS) [159], Meta-cognitive Neuro-fuzzy Inference System (McFIS) [160], Meta-cognitive Interval Type-2 Neuro-Fuzzy Inference System (McIT2FIS) [149] and SVM. Further, the performance of SRIT2NFIS is also compared with well-known type-2 algorithms including Type-2 Self-Organizing Neural Fuzzy System (T2SONFS), Mamdani-Type Reduced Interval Type-2 Neural Fuzzy System using Weighted Bound-set boundaries (MRIT2NFS-WB) and Interval Type-2 Neural Fuzzy Classifier learned through Soft Margin Minimization (IT2NFC-SMM) on benchmark data sets.

The rest of the chapter is organized as follows. In Section 4.1, a detailed review of existing type-2 systems followed by a review on self-regulation in Section 4.2 is presented.

In Section 4.3, the developed self-regulated interval type-2 neuro-fuzzy inference system is presented, followed by the performance evaluation in Section 4.4. In Section 4.5, the summary of this chapter is presented.

4.1 A Review of Interval Type-2 Neuro-Fuzzy Inference System literature

Fuzzy theory was first introduced by Zadeh in [161]. Fuzzy sets are characterized by a membership function which assigns a grade of membership between zero and one. In contrast to Boolean logic which can only take values zero and one, fuzzy logic takes a continuum of values between zero and one. Fuzzy logic defines intermediate values instead of defining the objects, completely true or completely false and as a result is able to mimic human-like thinking. It can handle concepts used in day to day life like "fast", "very fast", "slow", "very slow".

To handle uncertainties in data, Zadeh has proposed type-2 fuzzy sets in [121] and type-2 fuzzy logic systems employing these type-2 fuzzy sets have been studied in [122, 162]. Type-2 fuzzy sets allow researchers to model uncertainty, but it is computationally expensive. Interval type-2 fuzzy sets have been introduced to minimize the computational effort needed in type-2 fuzzy sets for handling data uncertainties [123].

There are two main approaches to design of a fuzzy logic system in the literature:

- Type-1 fuzzy sets: Membership functions are totally certain.
- Type-2 fuzzy sets: Membership functions are fuzzy which results in the uncertainty of rule antecedents and consequents.

4.1.1 Type-1 Fuzzy Sets

A type-1 fuzzy set, A , using a single variable, $x \in X$, can be represented as:

$$A = \{(x, \mu_A(x)) \mid \forall x \in X\} \quad (4.1)$$

A can also be defined as:

$$A = \int_{x \in X} \mu_A(x)/x \quad (4.2)$$

where, $\int$ denotes union over all admissible x .

A type-1 Gaussian membership function, $\mu_A(x)$ is a two-dimensional function that varies between zero and one for all $x \in X$ and is shown in Fig. 4.2. There exists a crisp membership value μ' for every input data point x' , hence type-1 membership functions are not able to handle uncertainty.

Figure 4.2: A Gaussian type-1 fuzzy membership function.

4.1.2 Type-2 Fuzzy Sets

A type-2 fuzzy set, $\tilde{A}$, is characterized by a type-2 membership function $\mu_{\tilde{A}}(x, u)$ [122] i.e.

$$\tilde{A} = \{((x, u), \mu_{\tilde{A}}(x, u)) \mid \forall x \in X, \forall u \in J_x \subseteq [0, 1]\} \quad (4.3)$$

where, $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$.

$\tilde{A}$ can also be defined as:

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x, u) / (x, u) \quad J_x \subseteq [0, 1] \} \quad (4.4)$$

where, $\int \int$ denotes union over all admissible x and u .

For each of the primary membership, there exists a corresponding secondary membership which can take values in the interval $[0,1]$. The use of type-2 fuzzy sets allow researchers to model uncertainty, but it is computationally expensive. Therefore, interval type-2 fuzzy sets have been introduced to minimize the computational effort needed in type-2 fuzzy sets [123].

4.1.3 Interval Type-2 Fuzzy Sets

When all $\mu_{\tilde{A}}(x, u)$ are equal to 1, then $\tilde{A}$ is called as an interval type-2 fuzzy set [123]. The Equation 4.4 might be defined for the interval type-2 fuzzy sets is given as:

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} 1 / (x, u) J_x \subseteq [0, 1] \} \quad (4.5)$$

where, $\int \int$ denotes union over all admissible x and u .

Usage of interval type-2 fuzzy sets is computationally cheap because the secondary membership value of the interval type-2 membership function is always equal to 1. The Footprint Of Uncertainty (FOU) of interval type-2 fuzzy set is shown in Fig. 4.3.

Based on these fuzzy sets, a class of algorithms known as Interval Type-2 Neuro-Fuzzy Inference Systems (IT2NFIS) has been developed [119, 124–136].

Interval type-2 neuro-fuzzy inference systems can broadly be classified based on several related criteria, as shown below:

- Classification based on the types of fuzzy inference mechanism
  - Takagi-Sugeno-Kang fuzzy inference mechanism
  - Mamdani fuzzy inference mechanism
- Classification based on the types of learning mode
  - Batch/ Offline learning
  - Sequential/ Online learning

Figure 4.3: Foot print of interval type-2 membership function modeled using uncertain mean

- Classification based on the types of uncertainty being modeled
  - Membership function with uncertain mean
  - Membership function with uncertain standard deviation
- Classification based on the types of consequent
  - Classifier type-Consequents are labels
  - Function approximation type-Consequents produce numerical values
- Classification based on the type-reduction mechanism
  - Karnik - Mendel algorithm
  - Others

4.1.4 Classification Based on the Types of Fuzzy Inference Mechanism

The two major inference mechanisms proposed in the literature of fuzzy inference systems are Takagi-Sugeno-Kang (TSK) fuzzy inference mechanism [163] and Mamdani fuzzy

inference mechanism [164].

4.1.4.1 Mamdani Fuzzy Inference Mechanism

Mamdani fuzzy inference mechanism [164], was proposed by E. H. Mamdani in 1974. The main advantage of using the Mamdani fuzzy inference mechanism is its rule interpretability. However, it lacks accurate generalization ability. As per this inference mechanism, for any fuzzy system with m input features, n output features and K rules, the k^{th} rule for such a system, R_k , is expressed as:

$$\begin{aligned} R_k : & \text{IF } x_1 \text{ is } \tilde{A}_{1k} \text{ AND } x_2 \text{ is } \tilde{A}_{2k} \text{ AND } \cdots \text{ AND } x_m \text{ is } \tilde{A}_{mk} \\ & \text{THEN } y_j = \tilde{B}_{cjk} \end{aligned} \quad (4.6)$$

where, $\mathbf{x} = [x_1, \cdots, x_m] \in \mathfrak{R}^m$ is the m -dimensional input features and $\mathbf{y} = [y_1, \cdots, y_n] \in \mathfrak{R}^n$ is the n -dimensional output. $\tilde{\mathbf{A}}_k = [\tilde{A}_{1k}, \cdots, \tilde{A}_{mk}] \in \mathfrak{R}^m$ are the interval type-2 fuzzy rule antecedents of k^{th} rule and $\tilde{B}_{cjk}$ is the corresponding interval type-2 fuzzy set consequent.

In a neuro-fuzzy inference system employing Mamdani fuzzy inference mechanism, first, the membership of the input vector with respect to existing rules in the network is calculated. The membership of i^{th} input feature of a sample $\mathbf{x}$ with respect to k^{th} rule is given by:

$$\phi_{ik} = \exp \left(\frac{(x_i - \mu_{ik})^2}{2\sigma_k^2} \right), \quad i = 1, \cdots, m \quad (4.7)$$

Here, μ_k and σ_k are the center and width of the k^{th} Gaussian rule. Other membership functions like triangular, trapezoidal membership functions can also be employed instead of using the Gaussian membership function for fuzzification. The next step is the aggregation of all the computed memberships. A minimum $T - norm$ operator is employed for feature aggregation, and it is given as:

$$\begin{aligned} \phi_K &= T - norm(\phi_{1k}, \cdots, \phi_{mk}) \\ &= \min(\phi_{1k}, \cdots, \phi_{mk}) \end{aligned} \quad (4.8)$$

A maximum $S - norm$ is used to compute the membership of the aggregated rule antecedents to the output. Techniques such as weighted average defuzzification, center of gravity defuzzification are used to compute the output of a Mamdani fuzzy inference

system. Many works are available in the literature of interval type-2 neuro-fuzzy inference systems that employ a Mamdani fuzzy inference mechanism [138, 143, 165].

4.1.4.2 Takagi-Sugeno-Kang Fuzzy Inference Mechanism

Takagi-Sugeno-Kang type fuzzy inference mechanism was first employed in [163]. Let us assume that the system consists of m input features, n output features and has K rules. The k^{th} rule for such a system, R_k , is expressed as:

$$\begin{aligned} R_k : & \text{IF } x_1 \text{ is } \tilde{A}_{1k} \text{ AND } x_2 \text{ is } \tilde{A}_{2k} \text{ AND } \cdots \text{ AND } x_m \text{ is } \tilde{A}_{mk} \\ & \text{THEN } y_j = \tilde{\alpha}_{c0jk} + \tilde{\alpha}_{c1jk}x_1 + \cdots + \tilde{\alpha}_{cnjk}x_n \end{aligned} \quad (4.9)$$

where, $\mathbf{x} = [x_1, \cdots, x_m] \in \mathfrak{R}^m$ are the m -dimensional input features and $\mathbf{y} = [y_1, \cdots, y_n] \in \mathfrak{R}^n$ are the n -dimensional output. $\tilde{\mathbf{A}}_k = [\tilde{A}_{1k}, \cdots, \tilde{A}_{mk}] \in \mathfrak{R}^m$ are the interval type-2 fuzzy rule antecedents of k^{th} rule and $\tilde{\alpha}_{ck} = [\tilde{\alpha}_{c0jk}, \cdots, \tilde{\alpha}_{cnjk}]$ are interval sets consequent parameters of the k^{th} rule, and

$$\tilde{\alpha}_{cljk} = [c_{ljk} - s_{ljk}, c_{ljk} + s_{ljk}], \quad l = 1, \cdots, p \quad (4.10)$$

In the fuzzy inference systems employing TSK, the overall output is computed as a weighted linear combination of inputs. This system has the capability of describing a non-linear system with a lesser number of rules. In such a neuro-fuzzy inference system, first, the membership of the input vector with respect to existing rules in the network is calculated. The membership of i^{th} input feature of a sample $\mathbf{x}$ with respect to k^{th} rule is given as:

$$\phi_{ik} = \exp\left(\frac{(x_i - \mu_{ik})^2}{2\sigma_k^2}\right), \quad i = 1, \cdots, m \quad (4.11)$$

Here, μ_k and σ_k are the center and width of the k^{th} Gaussian rule employed. Other radial basis function can also be used to fuzzify the inputs.

In the next step, the firing strength of a rule is calculated by the fuzzy meet operation using an algebraic product of membership functions of rule antecedents. It is given as:

$$\begin{aligned} \phi_k &= T - norm(\phi_{1k}, \cdots, \phi_{mk}) \\ &= \phi_{1k} \times \cdots \times \phi_{mk}. \end{aligned} \quad (4.12)$$

The output of a TSK fuzzy system is a weighted center of gravity defuzzified output, given as:

$$\hat{y} = \frac{\sum_{k=1}^K v_k \cdot \phi_k}{\sum_{k=1}^K \phi_k}. \quad (4.13)$$

In the literature of interval type-2 neuro-fuzzy inference systems, many works are available that employ a TSK fuzzy inference mechanism [119, 137, 141].

4.1.5 Classification Based on the Types of Learning Modes

In the previous section, the interval type-2 literature was classified on the basis of the type of fuzzy inference mechanism being employed. In this section, the literature has been classified on the basis of types of learning modes being employed. There are two commonly used learning modes in the literature, viz. batch/offline learning mode and sequential/online learning mode.

4.1.5.1 Batch/ Offline Learning Mode

In offline learning algorithms, it is assumed that all the training samples are available *a priori*. In batch learning mode, samples are assumed to arrive batch by batch and the network parameters are adapted based on it. For the sake of convenience, all the batch algorithms are also assumed to be in the offline category. In offline learning, the structure of the network is fixed and the rules are often initialized employing offline clustering algorithms. In the literature many algorithms employ an offline learning mode in type-2 fuzzy systems [124, 125, 129, 166–170].

In [124], IT2NFIS architecture with TSK based inference mechanism has been proposed. Here, gradient descent algorithm is employed to find the best parameters for a type-1 network, which are then used to construct a type-2 network. The parameters of this network are optimized using a search based technique. However, the optimal parameters for a type-1 system cannot fully exhibit the uncertainty handling property of a type-2 system. In [166], gradient descent based approach was directly employed to tune the parameters of an IT2NFIS.

In [129], three different architectures of Interval Type-2 Fuzzy Neural Network (IT2FNN) have been proposed. The major differences among all three architectures are the ways

in which antecedents and consequents have been integrated. The first architecture is an adaptive neural network with five layers. It uses only adaptive nodes and employs Karnik-Mendel algorithm for type-reduction. The second architecture is a six-layered interval type-2 fuzzy neural network. It employs interval type-2 neurons for antecedents and type-1 neurons for consequent. The third architecture is a seven-layered interval type-2 fuzzy neural network. It is similar to the second architecture, except that the type-reduction is integrated in the adaptive network layers. In the above mentioned architectures, the structure of IT2FNN is assumed to be fixed. The steepest distance algorithm is employed to determine the antecedent parameters as well as the consequent parameters. In [171], an extended Kalman filter based learning algorithm for type-2 fuzzy logic systems has been proposed. It has a fixed structure and employs an extended Kalman filter for tuning the parameters of the network.

In the literature, various other algorithms have been proposed [125, 167], where a gradient descent based learning approach is employed to tune the parameters of type-2 systems and a genetic algorithm based approach is used to search the accurate spread of uncertainties. In [125], an interval type-2 fuzzy neural network has been proposed that employs a genetic algorithm to tune the mean and deviation in the IT2FNN. In this system, the structure of interval type-2 network is assumed to be fixed rendering the algorithm offline. Following this work, various works have been proposed that employ different variants of evolutionary algorithms to tune the parameters of the network. In [167], a type-2 fuzzy neural network has been proposed that uses genetic algorithm based method for tuning the parameters of the network. Here, type-2 fuzzy sets handle uncertainties and the genetic algorithm based tuning achieves better classification results. In [128], an IT2FNN has been presented that employs interval type-2 fuzzy sets as antecedents and coefficients of the polynomial functions as the consequents of the rules. It employs a real-coded genetic algorithm for tuning the parameters of the network. In [168], a type-2 fuzzy neural network has been presented that employs a differential evolution based clustering for generating the fuzzy rule base. Differential evolution algorithm is a population based algorithm which is designed for numerical optimization problems. It implements a global search and has good convergence properties in multi-dimensional search spaces. A detailed review of evolutionary algorithms for type-2 systems can be found in [172].

Another category of work employed fuzzy clustering for structure learning [169, 170, 173]. In [170], a data-driven interval type-2 neuro-fuzzy inference system has been presented which aims to improve the model interpretability. Interval type-2 fuzzy sets have been used in its antecedent part and the consequent is of TSK type. A self-splitting clustering algorithm has been employed to generate the initial rule base. It employs a two-phase parameter-learning algorithm. A new cost function that takes into account both accuracy and transparent fuzzy set partition is defined in the first phase. Gradient descent and rule-ordered recursive least squares algorithms are employed to learn antecedent and consequent parameters so that cost function can be minimized. In the second phase, fuzzy sets are merged and then the consequent parameters are tuned to improve accuracy. In this algorithm, all the samples are used in both the phases rendering the algorithm offline. In [169], the design of a novel type 2 TSK-based structure for identification and control of dynamic plants has been presented. It employs the fuzzy clustering approach for structure learning. Fuzzy clustering and gradient learning algorithms are used for updating the parameters of the network. Clustering aims to identify data points from a large pool of data such that the system is represented in a concise manner. Cluster centers are then translated into fuzzy rules. After identification of rule antecedents by fuzzy clustering, the gradient descent algorithm is used to update the consequent of the fuzzy rules. It needs all the data samples before applying the fuzzy clustering, making the algorithm offline.

4.1.5.2 Sequential/ Online Learning Mode

In the previous section, a review of offline/ batch algorithms is presented. These approaches, however, cannot effectively handle temporally varying data due to the fixed network structure. To capture the temporally varying data, the network should evolve both, its structure and parameters based on the data. This category of algorithms is called sequential/ online algorithms. An online algorithm processes data samples one-by-one in a serial fashion, i.e., in the order in which samples are fed to the system without having the knowledge of all the samples *apriori*. This section also presents a review of the evolving and adaptive algorithms for IT2NFIS in which rules are grown and adapted on the basis of data. In the case of evolving and adaptive algorithms, the network is

initially assumed to contain zero rules, and the rules are grown and adapted based on the training data.

Initially, various algorithms were proposed that employed gradient descent for learning of the network [141, 143, 165, 174, 175]. In [141], a Simplified IT2FNN (SIT2FNN) has been proposed which uses gradient descent to tune the antecedent as well consequent parameters. SIT2FNN is self-evolving as it can grow the required number of rules and parameters automatically. It employs the design factors q_l and q_r that enable the adjustment of upper and lower values in the type-reduction step. This helps in overcoming the time required for arranging the consequent parameters. In [174], a Reduced Interval Type-2 Neural Fuzzy System using Weighted Bound-set boundaries (RIT2NFS-WB) has been proposed which employs online structure learning and gradient descent based parameter learning. In the structure part, it employs weights of the lower and the upper firing strength of rules to compute y_l and y_r . Due to this simplified type-reduction step, RIT2NFS-WB can be implemented on a chip for real-time applications. Here, the highly overlapping sets are merged to reduce the number of fuzzy sets and improve interpretability.

A Mamdani fuzzy inference mechanism for an evolving Interval Type-2 Fuzzy Inference System (eT2FIS) has been proposed in [143]. When a data sample arrives that contains new knowledge with respect to current knowledge, a new rule is generated. Rules are pruned from the system on becoming become irrelevant and the highly overlapping fuzzy labels are merged in the input-output spaces. A gradient descent based approach is used to update the centers and the width of the rules. However, due to the use of the Mamdani fuzzy inference mechanism, the system is not able to achieve good accuracy. A Mamdani-type Interval Type-2 Neural Fuzzy Chip with On-chip Incremental Learning ability (IT2NFC-OL) has been presented in [165]. Due to the use of iterative procedure by traditional interval type-2 fuzzy systems, finding system output for hardware implementation becomes computationally expensive. In IT2NFC-OL, a simplified type-reduction operation is presented to reduce the implementation cost. It uses gradient descent algorithm for parameter learning. The learning model is also implemented in a field-programmable gate array chip.

Another category of algorithms employed Kalman filters with gradient descent to tune the parameters of network [137–139, 176, 177]. A TSK fuzzy inference mechanism

for Self Evolving Interval Type-2 Fuzzy Neural Network (SEIT2FNN) has been proposed in [137]. SEIT2FNN employs an online clustering method to generate rules such that the input space is partitioned. It avoids generation of highly overlapping fuzzy sets in each input variable using a pre-specified threshold value. It uses a rule ordered Kalman filter for consequent and standard gradient descent for antecedent parameter learning. Initially, there are no rules in the SEIT2FNN. Rules are generated by using an online clustering method so that the input space is partitioned. The parameter learning phase occurs concurrently with the structure learning phase. All of the SEIT2FNN parameters are tuned for each piece of the incoming data. However, it requires all the samples to be presented multiple times, hence, it is not truly online. In [138], a Type-2 Self-Organizing Neural Fuzzy System (T2SONFS) and its hardware implementation has been presented. Interval type-2 fuzzy sets are used in the antecedent part and the consequent part is of Mamdani type. T2SONFS employs an online clustering algorithm to generate rules for structure identification. It employs gradient descent for antecedent parameter learning and a rule-ordered Kalman filter algorithm is used for tuning the consequent part parameters. T2SONFS is also implemented in hardware in a field-programmable gate array chip so that the functionality of circuits can be verified. In [177], a TSK-Type-Based Self-Evolving Compensatory Interval Type-2 Fuzzy Neural Network (TSCIT2FNN) has been presented. The firing strength of a rule is used to determine the generation of a new rule. It employs an online type-2 fuzzy clustering for deriving the rules and a variable-expansive Kalman filter algorithm is used for learning the consequent parameters. A gradient descent algorithm is employed to learn the antecedent type-2 fuzzy sets and compensatory weights.

In [176], a novel recurrent type-2 neural fuzzy system called Mutually Recurrent Interval Type-2 Neural Fuzzy System (MReIT2NFS) has been presented to identify time-varying systems. A local internal feedback and interaction loop is formed using the antecedent part by using the firing strength of the rule as a feedback to other rules including itself. The rule firing strength is used as criteria to identify, whether a new rule should be generated or not. The parameters are learned using a rule-ordered Kalman filter algorithm. A gradient descent algorithm is employed to learn the type-2 fuzzy sets used in the antecedent and the weights that represent the mutual feedback. The

insignificant rules are pruned based on the magnitude of recurrent weight. A Recurrent Self-Evolving Interval Type-2 Fuzzy Neural Network (RSEIT2FNN) for dynamic system processing has been presented in [139]. Here, the noise resistance is increased by using type-2 fuzzy sets in a recurrent neural fuzzy system. Interval type-2 fuzzy sets are used as antecedents in each of the recurrent fuzzy rules and the consequent is of the TSK type with interval weights. Online structure and parameter learning are used to learn rules in RSEIT2FNN. A rule-ordered Kalman filter algorithm is used for the parameter learning and a gradient descent algorithm is used to learn the antecedent type-2 fuzzy sets and internal feedback loop weights. However, it takes a lot of epochs for convergence.

Evolutionary algorithms have also been employed in the literature to learn the parameters of interval type-2 systems [134, 135]. In [134], a new reinforcement-learning method using Online Rule Generation and Q-value-aided Ant Colony Optimization (ORGQACO) for designing fuzzy controllers has been presented. The initial rule base is empty in ORGQACO. The parameter learning and structure learning occur simultaneously in ORGQACO. An online interval type-2 rule generation method is employed for structure evolution and partitioning the input space. The reinforcement local-global ant colony optimization algorithm and Q-values are used for designing the consequent part parameters. It uses ant pheromone trails and Q-values to select the consequent part. In [135], a reinforcement self-organizing interval type-2 fuzzy system with ant colony optimization has been presented. Here, the consequent part of each fuzzy rule is designed using ant colony optimization.

Another set of algorithms used support vector machines for learning parameters [119, 136]. In [119], an Interval Type-2 Fuzzy-Neural Network with Support Vector Regression (IT2FNN-SVR) has been proposed. IT2FNN-SVR learning consists of both structure learning and parameter learning. The structure learning algorithm is used for online rule generation. Here, even though the structure is learnt in an online fashion, two phases are required to learn the parameters which make the algorithm not truly online in nature. In [136], an IT2NFIS with support vector learning has been proposed for human posture classification. The structure identification proceeds via online clustering, while the parameters are learnt using support vector machine learning.

4.1.6 Classification Based on the Type of Uncertainty being Modeled

In the earlier sections, a review of the literature has been provided on the basis of type of fuzzy inference mechanism and the type of learning modes. In this section, a review of major membership functions employed in the literature of interval type-2 fuzzy inference systems is provided. The two types of membership function used to model uncertainty are Gaussian membership function with uncertain mean and Gaussian membership function with uncertain standard deviation.

4.1.6.1 Membership Function with Uncertain Mean

The Gaussian primary membership function having a fixed standard deviation and an uncertain mean that takes on values in $[\mu^l, \mu^r]$ is given as:

$$\phi(\mu_{ki}, \sigma_k, u_i) = \exp\left(-\frac{(u_i - \mu_{ki})^2}{2\sigma_k^2}\right) \quad (4.14)$$

where, $\mu_{ki} \in [\mu_{ki}^l, \mu_{ki}^r]$ and σ_k are the center and width of i -th feature of the k -th rule, respectively.

The Footprint Of Uncertainty (FOU) for Gaussian type-2 fuzzy sets with uncertain mean is shown in Fig. 4.4. The uniformly shaded FOU indicates that uniform weighting is assumed. In the literature, various fuzzy inference systems have successfully employed these membership functions [137, 141, 141, 143, 165, 170, 174].

4.1.6.2 Membership Function with Uncertain Standard Deviation

The Gaussian primary membership function having a fixed mean, μ , and an uncertain standard deviation takes on values in $[\sigma^l, \sigma^r]$ is given as:

$$\phi(\mu_{ki}, \sigma_k, u_i) = \exp\left(-\frac{(u_i - \mu_{ki})^2}{2\sigma_k^2}\right) \quad (4.15)$$

where, $\sigma_k \in [\sigma_k^l, \sigma_k^r]$ and μ_{ki} are width and center of i -th feature of the k -th rule, respectively.

The FOU for Gaussian type-2 fuzzy sets with uncertain standard deviation is shown in Fig. 4.5. The uniform shading over entire FOU means that uniform weighting is

Figure 4.4: Footprint of interval type-2 membership function modeled using uncertain mean.

assumed. In the literature, various fuzzy inference systems have successfully employed these membership functions [129, 154, 170, 178].

4.1.7 Classification Based on the Types of Consequent

In this section, the literature has been classified on the basis of types of consequent employed in interval type-2 fuzzy inference systems. They can be of two types: consequents are labels and consequents produce numerical values.

4.1.7.1 Function Approximation Type

Majority of interval type-2 neuro-fuzzy inference systems [137–139, 141, 143, 165, 174] produce numerical values as the output. In these systems, the consequents produce numerical values which are used for function approximation or system identification. Here, the cost function is defined on the basis of the difference between the output of the fuzzy inference system and the measured output of the system.

Figure 4.5: Interval type-2 fuzzy set with uncertain standard deviation and its foot print of uncertainty.

4.1.7.2 Classifier Type

These interval type-2 fuzzy inference systems [149, 175, 179] produce class labels as output. Here, the cost function is defined as the difference between the class number of an input with respect to its real class. In [175], an evolving type-2 fuzzy classifier has been presented which constructs an evolving working principle in the framework of interval type-2 fuzzy system. It is driven by a generalized interval type-2 fuzzy rule, where the premise part is composed of the multivariate Gaussian function with an uncertain non-diagonal covariance matrix, while employing a subset of the nonlinear Chebyshev polynomial as the rule consequents.

4.1.8 Classification Based on the Type-Reduction Mechanism

In this section, a brief review of the type-reduction mechanisms employed in the literature of interval type-2 fuzzy inference systems is presented. In the majority of the IT2NFIS, the Karnik-Mendel algorithm is used for type-reduction in the output processing layer.

4.1.8.1 Karnik - Mendel Algorithm

Majority of interval type-2 neuro-fuzzy inference systems [124, 125, 127, 137, 139] use the Karnik-Mendel (K-M) algorithm [27] in the output processing layer to calculate an interval type-1 set $[w_l, w_r]$. In the K-M iterative procedure, the consequent parameters are reordered in ascending order, and hence the firing strengths are also reordered accordingly, making the algorithm computationally intensive. The outputs y_l and y_r can be computed as follows:

$$y_l = \frac{\sum_{i=1}^{K'} F_i^{up} w_l^i + \sum_{i=K'+1}^K F_i^{lo} w_l^i}{\sum_{i=1}^{K'} F_i^{up} + \sum_{i=K'+1}^K F_i^{lo}} \quad (4.16)$$

and

$$y_r = \frac{\sum_{i=1}^{K''} F_i^{lo} w_r^i + \sum_{i=K''+1}^K F_i^{up} w_r^i}{\sum_{i=1}^{K''} F_i^{lo} + \sum_{i=K''+1}^K F_i^{up}} \quad (4.17)$$

where, w_l and w_r are consequent parameters, F^{up} and F^{lo} are upper and lower firing strengths, indexes l and r represent mean left and right limits, respectively.

4.1.8.2 Others

In SIT2FNN [141], a simplified type-reduction approach has been presented that involves adaptive adjustment of the upper and the lower values without using the K-M iterative procedure. It uses the design factors q_l and q_r that enable the adjustment of upper and lower values in the type-reduction step. Using factors $[q_l, q_r]$ instead of K-M iterative procedure significantly reduces the computational cost.

In RIT2NFS-WB [174], an interval type-2 neural fuzzy system with cheap type-reduction has been presented. Instead of using the expensive K-M algorithm, it weights the lower and upper firing strength of rules to compute y_l and y_r . Due to the simplified type-reduction step, RIT2NFS-WB can be implemented on a chip for real-time applications which require faster inference and low power consumption. Also, the highly overlapped sets are merged to reduce the number of fuzzy sets and improve interpretability. In [169], q coefficient is used to adjust the proportion of the upper and lower bound in terms of the final output.

4.1.9 Summary of Interval Type-2 Neuro-fuzzy Inference Systems Discussed in Literature

A summary of the presented interval type-2 neuro-fuzzy inference systems has been provided in Table 4.1. The above mentioned learning algorithms assume that the training samples are uniformly distributed over feature space. In practical problems, the sequential data does not satisfy this assumption and hence, the performance will be influenced by the training sequence. All these algorithms lack the ability to self-regulate their own knowledge, which might result in over-training. Meta-cognition facilitates self-regulation and has been shown to be effective in evolving the architecture efficiently [149–154]. In the next section, a review on meta-cognitive based self-regulation in learning is presented.

Table 4.1: Summary of major interval type-2 algorithms discussed in literature review.

Algorithm	Inference Mechanism	Type of Membership Function	Learning Mode
Wang et al. [125]	TSK	UM	Offline (Genetic Algorithm)
Cai et al. [167]	TSK	UM	Offline (Genetic Algorithm)
Castro et al. [129]	TSK	US	Offline
Juang and Chen [170]	TSK	US	Offline (two phase learning)
Juang and Tsao [137]	TSK	UM	Online (online clustering + gradient descent + rule ordered kalman filter)
Juang and Tsao [138]	Mamdani	UM	Online (online clustering + gradient descent + rule ordered kalman filter)
Juang and Hsu [134]	TSK	US	Pseudo-online (online clustering + evolutionary)
Juang et al. [135]	TSK	UM	Pseudo-online (online clustering + evolutionary)
Juang et al. [139]	TSK	UM	Online (online clustering + gradient descent + rule ordered kalman filter)
Juang et al. [119]	TSK	UM	Pseudo-online (online clustering + multiple phase learning)
Lin et al. [176]	TSK	UM	Online (online clustering + gradient descent + rule ordered kalman filter)
Lin et al. [177]	TSK	UM	Online (online clustering + gradient descent + rule ordered kalman filter)
Tung et al. [143]	Mamdani	UM	Online (online clustering + gradient descent + rule ordered kalman filter)
Juang et al. [165]	Mamdani	UM	Online (online clustering + gradient descent + rule ordered kalman filter)
Juang and Juang [174]	TSK	UM	Online (online clustering + gradient descent)
Lin et al. [141]	TSK	UM	Online (gradient descent)
Juang and Hsu [134]	TSK	UM	Online (online clustering + gradient descent)
			Online (online clustering + gradient descent)
			Pseudo-online (online clustering + evolutionary)

(a) Uncertain Mean(UM) (b) Uncertain Standard deviation (US)

4.2 Self-Regulation in Learning

Recent studies in human learning have suggested that the learning process is effective when the learners adopt self-regulation in learning process using meta-cognition [180]. The term meta-cognition is defined in [181] as one's knowledge concerning one's own cognitive processes or anything related to them. Precisely, the learner should control the learning process by planning and selecting learning strategies and monitor their progress by analyzing the effectiveness of the learning strategies. When necessary, these strategies should be adapted appropriately. There are several meta-cognition models available in human physiology and a brief survey of various meta-cognition models are reported in [182]. Among the various models, the model proposed by Nelson and Narens in [1] is simple and clearly highlights the various actions in human meta-cognition as shown in Fig. 4.6. The model is analogous to the meta-cognition in human beings and comprises of two components, the cognitive component and the meta-cognitive component. The information flow from the cognitive component to meta-cognitive component is considered monitoring, while the information flow in the reverse direction is considered control. The information flowing from the meta-cognitive component to the cognitive component either changes the state of the cognitive component or changes the cognitive component itself. Monitoring informs the meta-cognitive component about the state of cognitive component, thus continuously updating the meta-cognitive component's model of cognitive component, including, no change in state. Meta-cognition based self-regulation present in human beings provides a means to address what-to-learn, when-to-learn and how-to-learn, i.e., the ability to identify the specific piece of required knowledge, judge when to start and stop learning by emphasizing best learning strategy.

Self-regulation based learning has been used in the literature to improve the performance of neuro-fuzzy inference systems [149, 151–154, 179, 183] and soft-computing systems [150, 184–188]. It has been shown that algorithms employing self-regulation have better generalization ability. In the next section, a self-regulated interval type-2 neuro-fuzzy inference system is proposed that has the ability to handle non-stationary data. Here, self-regulation is realized using meta-cognition.

Figure 4.6: Nelson and Narens model [1] of meta-cognition.

4.3 Self-Regulated Interval Type-2 Neuro-Fuzzy Inference System

In this section, a detailed description of Self-Regulated Interval Type-2 Neuro-Fuzzy Inference System (SRIT2NFIS) architecture and the learning algorithm is presented. Let the training data be $\{(\mathbf{x}^1, c^1), \dots, (\mathbf{x}^t, c^t), \dots\}$, where $\mathbf{x}^t = [x_1^t, \dots, x_m^t]^T \in \mathbb{R}^m$ is a m -dimensional input vector of t^{th} sample, $c^t \in [1, 2, \dots, n]$ is the corresponding class label and n represents the total number of distinct classes in the data set. The coded class label, $\mathbf{y}^t \in \mathbb{R}^n$ is given as,

$$y_j^t = \begin{cases} 1, & \text{If } j == c^t \\ -1, & \text{otherwise} \end{cases} \quad j = 1, 2, \dots, n. \quad (4.18)$$

The objective of the SRIT2NFIS classifier is to approximate the underlying functional relationship $(\mathbf{x}^t \rightarrow \mathbf{y}^t)$ between the input $(\mathbf{x}^t)$ and output $(\mathbf{y}^t)$. Next, the SRIT2NFIS architecture is described, followed by its self-regulatory learning algorithm.

4.3.1 Self-Regulated Interval Type-2 Neuro-Fuzzy Inference System Architecture

The SRIT2NFIS architecture is a five-layered modified interval type-2 neuro-fuzzy inference system that predicts the class label for the current sample. It is presented in Fig. 4.7. Let us assume that K rules have been added to the network after processing $t - 1$ input samples. The detailed description of each layer on presenting the t^{th} sample is as follows:

Layer 1-Input layer: The input layer consists of m nodes and passes the input directly to the fuzzification layer. The output of i^{th} node of the input layer is given as:

$$u_i^t = x_i^t; \quad i = 1, 2, \dots, m. \quad (4.19)$$

Figure 4.7: Structure of self-regulated interval type-2 neuro-fuzzy inference system.

Layer 2- Fuzzification layer: Each node in this layer captures the non-stationarity in input features as an uncertainty bound by employing an interval type-2 Gaussian membership function. The membership of i^{th} input feature with k^{th} rule is given by:

$$\phi_{ik}^t = \exp \left(-\frac{(u_i^t - \mu_{ik})^2}{2\sigma_k^2} \right) \equiv \phi(\mu_{ik}, \sigma_k, u_i^t) \quad (4.20)$$

where, $\mu_{ik} \in [\mu_{ik}^1, \mu_{ik}^2]$ are the left and right limits of the center and σ_k is the width of k^{th} rule.

The footprint of uncertainty of this membership function can be represented as a bounded interval in terms of upper membership function ϕ^{up} and lower membership function ϕ^{lo} , as given below:

$$\phi_{ik}^{up,t} = \begin{cases} \phi(\mu_{ik}^1, \sigma_k, u_i^t) & u_i^t < \mu_{ik}^1 \\ 1 & \mu_{ik}^1 \leq u_i^t \leq \mu_{ik}^2 \\ \phi(\mu_{ik}^2, \sigma_k, u_i^t) & u_i^t > \mu_{ik}^2 \end{cases} \quad (4.21)$$

$$\phi_{ik}^{lo,t} = \begin{cases} \phi(\mu_{ik}^2, \sigma_k, u_i^t) & u_i^t \leq \frac{(\mu_{ik}^1 + \mu_{ik}^2)}{2} \\ \phi(\mu_{ik}^1, \sigma_k, u_i^t) & u_i^t > \frac{(\mu_{ik}^1 + \mu_{ik}^2)}{2} \end{cases} \quad (4.22)$$

The output of each node can be represented by the interval $\Phi_{ik}^t = [\phi_{ik}^{lo,t}, \phi_{ik}^{up,t}]$.

Layer 3- Firing layer: Each node in this layer represents the firing strength of a rule. It is calculated by the algebraic product of membership values of antecedents of a rule. The output, F_k is an interval type-1 fuzzy set given by:

$$F_k^t = [F_k^{lo,t}, F_k^{up,t}]; \quad k = 1, \dots, K \quad (4.23)$$

where

$$F_k^{lo,t} = \prod_{i=1}^m \phi_{ik}^{lo,t} \quad \text{and} \quad F_k^{up,t} = \prod_{i=1}^m \phi_{ik}^{up,t}; \quad k = 1, \dots, K. \quad (4.24)$$

Layer 4- Type reduction layer: Each node in this layer converts the interval type-1 fuzzy set to a fuzzy number for a particular rule using the Nie-Tan type-reduction approach [28]. This approach is a closed form approximation of the well-known Karnik-Mendel algorithm [27]. The output of each node is given as:

$$F_k^t = \alpha F_k^{lo,t} + (1 - \alpha) F_k^{up,t}; \quad k = 1, \dots, K \quad (4.25)$$

where, α is the weighted measure of uncertainty and is set as 0.5.

Layer 5-Output layer: This layer computes the output of the network by the normalized weighted sum of firing strengths of rules and is given as:

$$\hat{y}_j^t = \frac{\sum_{k=1}^K w_{kj} F_k^t}{\sum_{p=1}^K F_p^t}; \quad j = 1, 2, \dots, n \quad (4.26)$$

where, w_{kj} is the output weight connecting k -th rule with the j^{th} output node. The predicted class label corresponding to $\hat{\mathbf{y}}^t$ is given by $\hat{c}^t$, and is defined as:

$$\hat{c}^t = \arg \max_{j=1,2,\dots,n} (\hat{y}_j^t). \quad (4.27)$$

Next, the self-regulatory learning algorithm to estimate the rules and associated parameters from the data is presented.

4.3.2 Self-Regulated Interval Type-2 Neuro-Fuzzy Inference System Learning Algorithm

The SRIT2NFIS learning algorithm employs a regularized form of the projection based learning algorithm to compute the weights of the network. The projection based learning algorithm has been first proposed in [29]. It minimizes the hinge-error to find the optimal weights of the network, however, minimizing only the hinge-error may lead to over-fitting. Hence, in this work, the sum of squared hinge-error is regularized for all samples with a regularizing factor.

The hinge-error is defined as:

$$J_t = \sum_{j=1}^n \begin{cases} \left(y_j^t - \sum_{k=1}^K w_{kj} \bar{F}_k^t \right)^2, & \text{if } y_j^t \hat{y}_j^t < 1 \\ 0, & \text{otherwise} \end{cases} \quad (4.28)$$

where

$$\bar{F}_k^t = \frac{F_k^t}{\sum_{p=1}^K F_p^t}; \quad k = 1, \dots, K. \quad (4.29)$$

The total energy function, $J(\mathbf{W})$ is given as:

$$J(\mathbf{W}) = \frac{1}{2} \sum_{i=1}^t J_i + \sum_{k=1}^K \lambda^2 \|w_{kj}\|^2. \quad (4.30)$$

It should be noted that $J(\mathbf{W})$ is regularized to avoid over-fitting. Here, $\lambda > 0$ is the coefficient of regularization. If λ is chosen close to 0, regularization will not have any effect on $J(\mathbf{W})$, that may result in over-fitting. Similarly, a λ value close to 1 will result in higher penalty to the weights that may lead to under-fitting. It has been set to 0.1. The inclusion of the weight norm reduces overfitting by adding a complexity penalty term to the objective function. It penalizes the larger values of weights as larger weights can lead to complex models which might overfit the data. The Eqn. 4.30 models the problem as a least squares problem while using regularization to further constrain the resulting solution. It improves the generalization ability of the model by constraining it at training time. The second term in Eqn. 4.30 is a L2 regularization term which penalizes larger weight values thereby reducing the effect of overfitting. More details can be found in [189].

The aim of regularized projection based learning algorithm is to estimate the optimal output weights ($\mathbf{W}^*$) such that the total energy of the network is minimized.

$$\mathbf{W}^* = \arg \min_{\mathbf{W}} J(\mathbf{W}). \quad (4.31)$$

The optimal output weights ($\mathbf{W}^*$) are obtained by equating the first order partial derivative of $J(\mathbf{W})$ with respect to output weight to zero.

$$\frac{\partial J(\mathbf{W})}{\partial w_{pj}} = 0; \quad p = 1, \dots, K; \quad j = 1, \dots, n$$

By equating the first order partial derivative to zero and re-arranging the terms,

$$\sum_{k=1}^K \sum_{i=1}^t \bar{F}_k^i \bar{F}_p^i w_{kj} + \lambda^2 w_{pj} = \sum_{i=1}^t \bar{F}_p^i y_j^i; \quad p = 1, \dots, K; \quad j = 1, \dots, n \quad (4.32)$$

which in turn is written as:

$$\mathbf{A}\mathbf{W} + \lambda^2\mathbf{W} = \mathbf{D} \quad (4.33)$$

where, the projection matrix $\mathbf{A} \in \mathbb{R}^{K \times K}$ is given by:

$$a_{kp} = \sum_{i=1}^t \bar{F}_k^i \bar{F}_p^i, \quad k = 1, \dots, K; \quad p = 1, \dots, K \quad (4.34)$$

and the output matrix $\mathbf{D} \in \Re^{K \times n}$ is:

$$d_{pj} = \sum_{i=1}^t \bar{F}_p^i y_j^i; \quad p = 1, \dots, K; \quad j = 1, \dots, n. \quad (4.35)$$

From eqn. (4.33), $\mathbf{W}^*$ are obtained as

$$\mathbf{W}^* = (\mathbf{A} + \lambda^2 \mathbf{I})^{-1} \mathbf{D}. \quad (4.36)$$

When a new sample $(\mathbf{x}^t, \mathbf{y}^t)$ is presented to the network, the learning algorithm monitors the knowledge in the network with respect to knowledge present in the sample. It decides whether the sample is used for rule growing/parameter update or discarded without learning or reserved to be considered for learning at a later stage. SRIT2NFIS employs the prediction error, spherical potential [151] and maximum rule contribution to measure the knowledge in the sample. Here, the three knowledge measures and the working principle of rule growing/parameter update, deletion of sample and sample reserve are discussed.

Absolute maximum hinge-error: The absolute maximum hinge-error E^t is employed in order to measure the prediction error of the current sample. It is given as:

$$E^t = \max_{j=1, \dots, n} |e_j^t| \quad (4.37)$$

where

$$e_j^t = \begin{cases} y_j^t - \hat{y}_j^t, & y_j^t \hat{y}_j^t < 1 \\ 0, & otherwise \end{cases} \quad j = 1, \dots, n. \quad (4.38)$$

Class-Specific Spherical potential: Class-specific spherical potential is the measure of the knowledge contained in a sample [184]. A higher value of spherical potential indicates that similar knowledge is present in the network and a lower value indicates that the sample contains novelty. It is defined as the average distance of the current sample from significantly contributing rules of the same class in a hyper-dimensional feature space.

$$\psi^t = \frac{1}{K_c} \sum_{k=1}^{K_c} F_k^t. \quad (4.39)$$

Here, K_c represents the number of significantly contributing rules belonging to class c . A rule is deemed to be contributing significantly if its membership $F_k^t > 0.01$.

Rule maximum contribution (F'): For a given sample, the maximum contribution is determined as $\max_{k=1,\dots,K} F_k$.

The rule growing criterion in [29] is only applicable in the case of radial basis networks where the final output is zero, when none of the neurons produce significant output. However, in case of neuro-fuzzy inference systems, the predicted output depends on the normalized firing strengths of the rules (ref. Eqn. (4.26)). Here, even though the rule produces zero firing strength, the normalized value is greater than zero and hence error may be smaller which might affect the learning ability. Hence, the rule growing criterion is modified as given below.

Rule growing criterion: A new rule is added to the network if the current sample contains significant knowledge, absolute maximum hinge-error for the sample is very high and the estimated class label is incorrect or no rule is fired for the sample even if the estimated class label is correct. The rule addition condition is given as:

$$\begin{aligned} & (c^t \neq \hat{c}^t \ \& \ E^t > E_a \ \& \ \psi^t < E_s \ \& \ F' < 0.001) \\ & \textbf{OR} \\ & (c^t = \hat{c}^t \ \& \ F' < 0.001) \end{aligned} \tag{4.40}$$

i.e. if the absolute maximum hinge-error is greater than the add threshold (E_a), and the spherical potential ψ^t is less than the novelty threshold (E_s) or the predicted output is equal to actual output but no rule is fired for the sample. The parameters, E_a and E_s have to be tuned for all the problems.

Here, E_a is self-adapted as:

$$E_a := (1 - \gamma)E_a + \gamma E^t. \tag{4.41}$$

The self-regulating add threshold, E_a is designed such that the global knowledge is captured first and at a later stage the network is fine tuned by virtue of its self-regulating nature. If its value is set too high, fewer new rules will be added to encapsulate the knowledge that will affect the generalization. If a small value is chosen, it will result in a large network size. The suggested range of this threshold is [1.01, 1.20] and is determined using a grid search with five-fold cross-validation.

Here, γ is the slope parameter that decides the slope at which E_a and the update threshold increases (Eqn. (4.55)), and is set to 0.99. It is chosen on the basis of the

number of samples present in a data set. For a large number of training samples, it is kept low to let the thresholds grow at a slower rate as many samples are present to reach the optimum value. For a lower number of training samples, it has to be kept high to allow the thresholds to reach an optimum value faster. Since, BCI problem considered in this work, has lower number of training samples, γ is kept constant at 0.99.

The novelty threshold, E_s is employed to decide how far a newly presented sample should be from the existing rule centers, in order to be added as a new rule. If this threshold is chosen closer to one (i.e., current sample can be close to existing rules in the projection space), many rules will be added to the network that might affect its generalization ability. On the other hand, if this threshold is chosen closer to zero, few rules will be added to the network, that might in turn affect the predictability of the network. The suggested range of this threshold is set as [0.01, 0.60]. A grid search with five-fold cross-validation is used to find this parameter.

For the newly added $K + 1^{th}$ rule, the center μ_{K+1}^l is initialized based on the current sample as:

$$\mu_{K+1}^l = [\mathbf{x}^t + 0.1, \mathbf{x}^t - 0.1] \quad (4.42)$$

where, the superscript l denotes the class of rule. The width of newly added rule is assigned by taking into consideration the distance from inter/intraclass rules. Let nrI be the nearest rule in the interclass ('other class') and nrS be the nearest rule in the intraclass ('same class') and are defined as:

$$\begin{aligned} nrS &= \arg \min_{l=c; \forall k} \|\mathbf{x}^t - \mu_k^l\| \\ nrI &= \arg \min_{l \neq c; \forall k} \|\mathbf{x}^t - \mu_k^l\|. \end{aligned} \quad (4.43)$$

Let the Euclidean distance between the current sample and the nearest intra/interclass rules be defined as:

$$d_S = \|\mathbf{x}^t - \mu_{nrS}\|, \quad d_I = \|\mathbf{x}^t - \mu_{nrI}\|. \quad (4.44)$$

When the current sample ($\mathbf{x}^t$) is nearer to the rule belonging to the same class, it does not overlap with the rules of other class. In this case, the width is assigned as:

$$\sigma_{K+1} = \kappa * d_S \quad (4.45)$$

where, κ controls the overlap between the newly added rule and the nearest rule in the same class. In case of BCI problems, it is desired to have a high overlap among the rules of the same class. Hence, κ is chosen as 0.95.

When the current sample ($\mathbf{x}^t$) is nearer to the rule belonging to interclass class, implying overlap of sample with rules of other class, the width is assigned as:

$$\sigma_{K+1} = \eta * d_I. \quad (4.46)$$

Interclass overlap factor (η) determines the overlap with the nearest rule in the interclass. Choosing a lower value of η will result in little overlap among the rules which will affect the network's generalization ability. Similarly, a larger overlap among the rules will also affect the performance adversely. The suitable range for this threshold is [0.1, 0.4] and is obtained using a grid search with five-fold cross-validation.

When a new rule is added based on the t^{th} sample, the projection matrix $\mathbf{A}$ is updated as,

$$\mathbf{A}_{(K+1) \times (K+1)} = \begin{bmatrix} \mathbf{A}_{K \times K} + (\bar{\mathbf{F}}^t)^T \bar{\mathbf{F}}^t & \mathbf{a}_{K+1}^T \\ \mathbf{a}_{K+1} & a_{K+1, K+1} \end{bmatrix} \quad (4.47)$$

where, $\mathbf{a}_{K+1} \in \mathbb{R}^{1 \times K}$ is

$$a_{K+1, p} = \sum_{h=1}^t \bar{\mathbf{F}}_{K+1}^h \bar{\mathbf{F}}_p^h, \quad p = 1, \dots, K \quad (4.48)$$

and

$$a_{K+1, K+1} = \sum_{h=1}^t \bar{\mathbf{F}}_{K+1}^h \bar{\mathbf{F}}_{K+1}^h. \quad (4.49)$$

The output matrix $\mathbf{D}$ is updated as,

$$\mathbf{D}_{(K+1) \times n} = \begin{bmatrix} \mathbf{D}_{K \times n}^{t-1} + (\bar{\mathbf{F}}^t)^T (\mathbf{y}^t)^T \\ \mathbf{d}_{K+1} \end{bmatrix} \quad (4.50)$$

and $\mathbf{d}_{K+1} \in \mathbb{R}^{1 \times n}$ is given as,

$$d_{K+1, j} = \sum_{h=1}^t \bar{\mathbf{F}}_{K+1}^h y_j^h, \quad j = 1, \dots, n. \quad (4.51)$$

The output weights are re-estimated as:

$$\mathbf{W}^* = (\mathbf{A} + \lambda^2 \mathbf{I})^{-1} \mathbf{D}. \quad (4.52)$$

The inverse of the matrix is calculated recursively using matrix identities [190] as shown below:

$$\begin{aligned}
 (\mathbf{A}_{(K+1) \times (K+1)}^t + \lambda^2 \mathbf{I})^{-1} &= \begin{bmatrix} (\mathbf{A}_{K \times K}^t + \lambda^2 \mathbf{I})^{-1} & 0 \\ 0 & 0 \end{bmatrix} \\
 + \frac{1}{\Delta} \begin{bmatrix} (\mathbf{A}_{K \times K}^t + \lambda^2 \mathbf{I})^{-1} \mathbf{a}^T \\ 1 \end{bmatrix} \begin{bmatrix} (\mathbf{A}_{K \times K}^t + \lambda^2 \mathbf{I})^{-1} \mathbf{a}^T \\ 1 \end{bmatrix}^T
 \end{aligned} \tag{4.53}$$

where,

$$\mathbf{A}_{K \times K}^t = \mathbf{A}^{\mathbf{t}-1} + \bar{F}^{\mathbf{t}T} \bar{F}^{\mathbf{t}}$$

$$\Delta = a_{k+1, K+1} - \mathbf{a}_{K+1} (\mathbf{A}_{K \times K}^{\mathbf{t}-1} + \lambda^2 \mathbf{I})^{-1} \mathbf{a}_{K+1}^T$$

and $(\mathbf{A}_{K \times K}^t + \lambda^2 \mathbf{I})^{-1}$ is calculated as:

$$(\mathbf{A}_{K \times K}^t + \lambda^2 \mathbf{I})^{-1} = (\mathbf{A}^{\mathbf{t}-1} + \lambda^2 \mathbf{I})^{-1} - \frac{(\mathbf{A}^{\mathbf{t}-1} + \lambda^2 \mathbf{I})^{-1} \bar{F}^{\mathbf{t}T} \bar{F}^{\mathbf{t}} (\mathbf{A}^{\mathbf{t}-1} + \lambda^2 \mathbf{I})^{-1}}{1 + \bar{F}^{\mathbf{t}} (\mathbf{A}^{\mathbf{t}-1} + \lambda^2 \mathbf{I})^{-1} \bar{F}^{\mathbf{t}T}}. \tag{4.54}$$

In [131, 157], alternate local approach for learning the consequent parameters are provided.

Parameter update criterion: The output weights of the network are updated if the estimated class label is correct and the absolute maximum hinge-error for the current sample is greater than the update threshold.

The update criterion is given as:

$$c^t == \hat{c}^t \ \& \ E^t > E_l. \tag{4.55}$$

Here, E_l is self-adaptive parameter update threshold. It is regulated based on the prediction error as:

$$E_l := (1 - \gamma)E_l + \gamma E^t. \tag{4.56}$$

This threshold decides the prediction error achieved before the parameters of the rules are updated using the regularized projection based learning algorithm. A lower value of this threshold results in more rules being updated, which might in turn affect the generalization ability of the network. On the other hand, a higher value of this threshold results in fewer rules being updated leading to the under performance of the network. Similar to the rule add threshold (E_a), parameter update threshold is also self-regulating in nature. One needs to choose this parameter in the range of [0.04, 0.2] using a grid search with five-fold cross-validation.

If t^{th} sample is used to update the parameters of network, the matrices $\mathbf{A} \in \Re^{K \times K}$ and $\mathbf{D} \in \Re^{K \times n}$ are updated as:

$$\mathbf{A} = \mathbf{A} + (\bar{F}^t)^T F^t \quad (4.57)$$

$$\mathbf{D} = \mathbf{D} + (\bar{F}^t)^T (\mathbf{y}^t)^T. \quad (4.58)$$

When a sample is used to update the output weight, the regularized projection based learning algorithm updates the weight parameters as follows:

$$\frac{\partial J(\mathbf{W}^t)}{\partial w_{pj}} = \frac{\partial J(\mathbf{W}^{t-1})}{\partial w_{pj}} + \frac{\partial J_t(\mathbf{W}^t)}{\partial w_{pj}} = 0 \quad (4.59)$$

where $p = 1, \dots, K; j = 1, \dots, n$.

Equating partial derivative to zero and re-arranging Eqn. (4.59)

$$(\mathbf{A}^{t-1} + (\bar{F}^t)^T \bar{F}^t + \lambda^2) \mathbf{W}^t - (\mathbf{D}^{t-1} + (\bar{F}^t)^T (\mathbf{y}^t)^T) = 0. \quad (4.60)$$

By substituting $\mathbf{D}^{t-1} = \mathbf{A}^{t-1} \mathbf{W}^{t-1} + \lambda^2 \mathbf{W}^{t-1}$ and $\mathbf{A}^t = \mathbf{A}^{t-1} + (\bar{F}^t)^T \bar{F}^t$ and adding and subtracting $(\bar{F}^t)^T \bar{F}^t \mathbf{W}^{t-1}$ and $\lambda^2 \mathbf{W}^{t-1}$, the output weights are updated as:

$$\mathbf{W} = \mathbf{W} + (\mathbf{A} + \lambda^2 \mathbf{I})^{-1} (\bar{F}^t)^T (\mathbf{e}^t)^T. \quad (4.61)$$

Rule Pruning Criterion: A rule is insignificant if the contribution of the rule (β_k) in the same class is less than a pruning threshold E_p for E_w consequent samples in the same class. This rule can be removed from the network. The contribution of a rule (β_k) is calculated as:

$$\beta_k = \frac{E^t}{\psi_c^t} \bar{F}_k^t \max |\mathbf{w}_k|. \quad (4.62)$$

The pruning threshold is chosen on the basis of the contribution of a particular rule. If the contribution of the rule is lower for a pruning window, E_w , the particular rule does not affect the output of the network and should be removed from the network. It is set as 0.01. This means that if the contribution of the particular rule is lower than 0.01 percent, its effect is minimal. Hence, this rule can be removed from the network. The pruning window, E_w is chosen on the basis of the number of samples present in the data set. In case of BCI problems where the data set is small, if the contribution of a particular rule

is very low for nearly 5-6 % of samples consecutively, it is removed from the network. Hence, E_w is set as 10.

Sample deletion criterion: A sample is deleted if similar knowledge is already present in the network. The sample deletion condition is given as:

$$c^t == \hat{c}^t \ \& \ E^t < E_d \quad (4.63)$$

i.e. if the prediction error for the current sample is less than delete threshold and the predicted class label is same as the actual class label, the knowledge contained in the sample is already present in the network. Hence, the sample is not learnt and discarded. Here, E_d is the sample delete threshold. It prevents the network to learn similar samples, thereby avoiding over-training and reducing the computational effort. It is chosen based on the desired confidence on the correctness of the classification of a sample. It is fixed at 0.05 for all the problems. This means that if a sample is classified correctly with 95 percent confidence level, that particular sample gets deleted. If the delete threshold is kept too high (lower confidence), too many samples will be deleted and if it is kept too low (high confidence), no sample will get deleted. Hence, E_d is chosen as 0.05 (based on 95 percent confidence level) for all the problems.

Sample reserve criterion: If the current sample does not satisfy rule growing, parameter update or sample deletion criterion, it is pushed to the rear-end of the training data stream. The self-regulating nature of the addition and learning thresholds facilitate the usage of these samples at a later stage.

These strategies are repeated for every sample in the network. In comparison to the proposed algorithms in the literature, the self-regulating nature of SRIT2NFIS enables the network to learn and generalize the knowledge, efficiently. The SRIT2NFIS algorithm is also summarized using a Flowchart 4.8. In the next section, a detailed performance evaluation of SRIT2NFIS is presented.

4.4 Performance Evaluation

In the previous section, the principle and working of the self-regulated interval type-2 neuro-fuzzy inference system was discussed in detail. In this section, the performance of SRIT2NFIS is evaluated using benchmark data sets from the UCI machine learning repository and compared with type-1 and type-2 algorithms.

Figure 4.8: Flowchart for SRIT2NFIS.

4.4.1 Performance Evaluation of SRIT2NFIS using Benchmark Data Sets

The performance of SRIT2NFIS is evaluated on a set of benchmark classification problems from the UCI machine learning repository [191]. The performance of SRIT2NFIS is compared with an Evolving Fuzzy Rule based Classifier (eClass) [157], On-line Sequential Fuzzy Extreme Learning Machine (OS-F-ELM) [158], Sequential Adaptive Fuzzy Inference System (SAFIS) [159], Meta-cognitive Neuro-fuzzy Inference System (McFIS) [160], Meta-cognitive Interval Type-2 Neuro-Fuzzy Inference System (McIT2FIS) [149]

Table 4.2: Classification data set

Data set	No. of features	No. of classes	Samples	
			Train	Test
PIMA	8	2	400	368
BC	9	2	300	383
ION	34	2	100	251
Iris	4	3	45	105
IS	19	7	210	2100

and SVM. The algorithms used in comparison, eClass, OS-F-ELM, SAFIS, McFIS realize a Takagi-Sukeno-Kang type-1 fuzzy inference mechanism and McIT2FIS realizes a type-2 system.

Table 4.2 shows the detailed description (number of features, classes and size of training and testing data) of data sets used in the study, viz., PIMA Indian diabetes (PIMA), Breast Cancer (BC), Ionosphere (ION), Iris classification and Image Segmentation (IS). It should be noted that in BC, 16 samples in the data set contain missing attributes, and hence discarded.

In all the problems, the performance of all the algorithms is measured using overall accuracy (η_o), average accuracy (η_a). These measures are defined below:

$$\eta_o = \frac{1}{N} \times \sum_{j=1}^n q_{jj} \times 100\% \quad (4.64)$$

$$\eta_a = \frac{1}{N} \times \sum_{j=1}^n \frac{q_{jj}}{N_j} \times 100\% \quad (4.65)$$

where, N is the total number of samples, N_j and q_{jj} are the total number of samples and the number of correctly classified samples in class j .

The performance of each algorithm on the classification problems is given in Table 4.3. It can be seen from the table that SRIT2NFIS performs better than the compared algorithms with fewer number of rules. The performance comparison on binary classification problems (PIMA, BC, ION) clearly shows that the testing accuracy of SRIT2NFIS is approximately 5-6% better than the type-1 classification algorithms on an average. In

Table 4.3: Performance evaluation of SRIT2NFIS using benchmark data set

Data set	Algorithm	No. of Rules	CPU (Sec)	Testing Accuracy	
				overall (η_o)	average (η_a)
PIMA	SVM	285	-	75.77	72.52
	OS-F-ELM	100	-	68.75	66.36
	eClass	13	-	66.3	61.13
	SAFIS	179	-	75.82	71.46
	McFIS	133	-	75.81	75.65
	McIT2FIS	61	-	74.94	74.31
	SRIT2NFIS	40	0.22	78.53	76.13
BC	SVM	92	-	96.34	95.24
	OS-F-ELM	90	-	91.64	89.45
	eClass	8	-	75.19	76.62
	SAFIS	55	-	97.39	97.85
	McFIS	33	-	97.29	97.85
	McIT2FIS	8	-	97.12	97.65
	SRIT2NFIS	6	0.08	97.39	97.65
ION	SVM	43	-	91.21	85.41
	OS-F-ELM	50	-	77.69	71.34
	eClass	51	-	79.86	78.04
	SAFIS	46	-	91.63	91.78
	McFIS	23	-	94.20	93.29
	McIT2FIS	16	-	93.23	91.78
	SRIT2NFIS	11	0.09	95.22	94.31
Iris	SVM	25	-	96.19	96.19
	OS-F-ELM	25	-	97.14	97.14
	eClass	8	-	96.19	96.19
	SAFIS	24	-	92.38	92.38
	McFIS	6	-	97.14	97.14
	McIT2FIS	5	-	98.10	98.10
	SRIT2NFIS	5	0.10	98.10	98.10
IS	SVM	127	-	91.38	91.38
	OS-F-ELM	30	-	87.33	87.33
	eClass	87	-	90.70	90.70
	SAFIS	156	-	90.28	90.38
	McFIS	78	-	91.33	91.33
	McIT2FIS	37	-	92.29	92.29
	SRIT2NFIS	30	0.21	92.86	92.86

all these problems, SRIT2NFIS employs a significantly lesser number of rules compared to the type-1 algorithms. Similarly, compared to McIT2FIS, SRIT2NFIS achieves better accuracy for all the binary classification problem, using lesser number of rules. In case of multi-category classification problems (Iris, IS), SRIT2NFIS achieves better performance than the compared algorithms using lesser number of rules.

The major reasons for the improved performance of SRIT2NFIS could be attributed to the following reasons. 1) The presence of interval type-2 sets in SRIT2NFIS enables it to perform better than the compared type-1 algorithms. 2) Self-regulation in learning has helped the network to learn and generalize efficiently, using fewer number of rules. 3) The use of class-specific information in monitoring and in control has helped the classifier to learn efficiently. 4) The regularized projection based learning algorithm computed optimal weights and helps in avoiding over-fitting.

4.4.2 Comparison of SRIT2NFIS with Interval Type-2 Algorithms

The performance of SRIT2NFIS is compared with the other type-2 systems, including T2SONFS, MRIT2NFS-WB and IT2NFC-SMM using three binary classification problems (SVMguide 1, Heart-s and Tao). The SVMguide 1 data set consists of four attributes with 3089 samples for training and 4000 samples for testing. The Heart-s consists of 13 attributes and 270 samples and Tao data set consists of 2 attributes and 1888 samples. As suggested in the literature [136], ten-fold cross-validation for Heart-s and Tao data sets has been adapted for comparative study.

Table 4.4 compares the performance of SRIT2NFIS with existing type-2 systems (T2SONFS, MRIT2NFS-WB, IT2NFC-SMM) on the above mentioned data sets. All the results except SRIT2NFIS have been presented from [136]. It can be seen from the Table 4.4 that SRIT2NFIS performs better than these algorithms for all the three data sets thereby proving its effectiveness.

4.5 Summary

In this chapter, a self-regulated interval type-2 neuro-fuzzy inference system is presented to handle the non-stationarity in data. SRIT2NFIS handles the non-stationarity as un-

Table 4.4: Performance comparison of SRIT2NFIS with existing type-2 systems.

Data set	Algorithm	Rules	Testing Accuracy
SVMguide1	T2SONFS	4	85.0
	MRITNFS-WB	4	88.07
	IT2NFC-SMM	4	94.07
	SRIT2NFIS	4	95.97
Heart-s	T2SONFS	4	79.25 ± 6.24
	MRITNFS-WB	4	82.95 ± 6.01
	IT2NFC-SMM	4	86.07 ± 3.81
	SRIT2NFIS	4.1 ± 0.56	90.00 ± 4.29
Tao	T2SONFS	4	85.32 ± 3.58
	MRITNFS-WB	4	86.11 ± 3.34
	IT2NFC-SMM	4	96.79 ± 2.27
	SRIT2NFIS	4.3 ± 0.82	97.24 ± 0.86

certainty in the data distribution space using the interval type-2 fuzzy sets. The parameters of SRIT2NFIS are learnt using a self-regulated learning algorithm that evolves its structure automatically and updates the parameters based on the knowledge gap between the current sample and the knowledge in the network. A regularized projection based learning algorithm and a modified rule addition criterion has also been incorporated to improve the generalization performance of SRIT2NFIS. The performance of the proposed SRIT2NFIS has been evaluated using a number of benchmark binary as well as multi-category classification problems. The qualitative and quantitative performance analysis using multiple data sets clearly indicates the superior performance of the proposed SRIT2NFIS. Next, the SRIT2NFIS algorithm is combined with RoCSP robust feature extraction algorithm proposed in Chapter 3 to form an intelligent BCI system in the next chapter.

Chapter 5

An Intelligent-Brain Computer Interface for Handling Non-stationarities in EEG Signals

In the previous chapters, a Robust Common Spatial Pattern (RoCSP) algorithm for robust feature extraction and a Self-Regulated Interval Type-2 Neuro-Fuzzy Inference System (SRIT2NFIS) classifier capable handling of handling non-stationarity has been proposed. This chapter combines RoCSP and SRIT2NFIS to form an Intelligent-Brain Computer Interface (I-BCI) system (shown in Fig. 5.1) that has the capability to handle the non-stationarity in EEG data. It employs RoCSP for feature extraction and SRIT2NFIS for classification.

Figure 5.1: Brain signal processing framework.

A time series $\{x_t\}$ is called strictly stationary if for all $(t_1, \dots, t_n)$ the joint distributions of $x_{t_1}, \dots, x_{t_n}$ and those of $x_{t_1+h}, \dots, x_{t_n+h}$ coincide for all time shifts h . Intuitively, stationarity means that the data of two different epochs comes from the same distribution.

This property does not hold for brain signals such as EEG because different processes are active in the brain at different times and the sensory input also changes constantly. The objective of the I-BCI algorithm is to handle non-stationarity in EEG data and perform classification. The problem is tractable since I-BCI is a combination of RoCSP and SRIT2NFIS and the time complexity of both RoCSP and SRIT2NFIS (sequential) is in polynomial time. The performance of I-BCI in handling non-stationarity in various intra/inter-session settings is evaluated using two publicly available BCI competition data sets. First, the ability of I-BCI to handle inter-session non-stationarity across different days is evaluated using the data set-IIa (six classification tasks) from BCI competition IV [116]. Next, the performance of I-BCI is evaluated on the data set-IIIa [192] from BCI competition III [18] to check its capability to handle intra-session non-stationarity. Next, a detailed performance evaluation of I-BCI is presented.

5.1 Performance Evaluation of I-BCI

In general, in an intra-session setting, non-stationarity may arise from activities such as swallowing, yawning and change in neural processes during different runs. In an inter-session setting, it may arise from the measurement factors such as a change in positioning of electrodes in addition to the reasons mentioned above. Further, if the sessions are conducted on different days, additional factors like change in environment, change in sleep pattern and a different mental state of the subject may also cause non-stationarity [23]. To study the effectiveness of the proposed approach, different BCI benchmark competition data set [18, 116] are selected that encompass the above mentioned conditions. The performance of the proposed I-BCI algorithm is compared with other existing methods in the literature for the above mentioned scenarios.

First, the performance of I-BCI to handle non-stationarity in inter-session data is evaluated using the data set-IIa from the BCI competition IV [116]. Next, its performance is evaluated for an intra-session setting using the data set-IIIa [192] from the BCI competition III [18]. The EEG signals are band passed from 8 to 30 Hz using fifth-order Butterworth filter. Further, a discussion on the synergy of the hybrid and the contribution of individual components, RoCSP and SRIT2NFIS is provided.

5.1.1 Performance of I-BCI in Handling Intersession Non-stationarity - BCI Competition IV Data Set-IIa

The performance of I-BCI for the six classification tasks is presented in Table 5.1. It can be clearly seen from the Table 5.1 that I-BCI outperforms the baseline CSP algorithm for all the classification tasks. Results show that it is more effective on the subjects with poor performance for the CSP algorithm (e.g., subject A2 in classification tasks left vs right, left vs tongue and foot vs tongue). On an average, the I-BCI algorithm outperforms the CSP algorithm by 7%. It achieves the highest classification accuracy of 90.97% for left vs tongue task followed by right vs foot task.

Table 5.2 shows the average performance of the CSP algorithm, its variants, and I-BCI for all the six classification tasks. The performance of I-BCI is compared with algorithms that regularize the CSP algorithm (GLRCSP, CCSP1, CCSP2, DLCSPauto, DLCSPcv, DLCSPcvdiff, SSRCSPP, TRCSP, WTRCSP and SRCSP) to handle non-stationarity. It can be observed from Table 5.2 that I-BCI outperforms all the above mentioned algorithms for all six classification tasks. It achieves the highest improvement of 8-10% in performance over all the other algorithms for the complex task like foot vs tongue task. The improvement in performance achieved by I-BCI may be due to its ability to handle the non-stationarity as uncertainty.

Table 5.1: Classification performance (Subject Mean(SM) in % and Standard Deviation(SD)) obtained for six classification tasks by using CSP-LDA and I-BCI algorithm for the BCI competition IV data set-IIa.

Subj ects	Six motor-imagery tasks of the BCI competition IV data set-IIa											
	Left vs Right		Left vs Foot		Left vs Tong.		Right vs Foot		Right vs Tong.		Foot vs Tong.	
	CSP	I-BCI	CSP	I-BCI	CSP	I-BCI	CSP	I-BCI	CSP	I-BCI	CSP	I-BCI
A1	88.89	93.75	97.92	100	96.53	100	99.31	100	100	100	72.22	77.78
A2	51.39	71.53	69.44	81.25	67.36	73.61	79.86	86.11	63.89	73.61	70.83	82.64
A3	96.53	98.61	93.06	96.53	94.44	97.22	91.67	97.92	96.53	100	68.75	79.86
A4	70.14	76.39	75.69	93.75	90.28	93.06	88.19	92.36	84.72	90.97	55.55	78.47
A5	54.86	69.44	61.11	75.69	69.44	79.17	64.58	75.00	65.97	74.31	67.36	76.39
A6	71.53	74.31	56.94	74.31	71.53	80.56	63.19	72.92	68.75	74.31	66.67	79.86
A7	81.25	83.33	98.61	100	97.22	98.61	99.31	99.31	97.22	97.92	81.94	86.81
A8	93.75	97.92	86.81	90.97	90.28	97.22	88.88	93.75	90.97	92.36	80.56	90.97
A9	93.75	94.44	95.14	97.92	95.14	99.31	81.94	93.75	86.11	93.75	90.97	96.53
SM	78.01	84.41	81.64	90.05	85.80	90.97	84.10	90.12	83.80	88.58	72.76	83.26
SD	17.01	11.88	16.23	10.29	12.54	10.25	13.23	10.09	14.16	11.32	10.36	6.82

Table 5.2: Comparison of classification performance (SM in % and SD) obtained for six classification tasks for CSP algorithm and its variants with the proposed I-BCI on the intersession (different days) data set II-a from BCI competition-IV

(a) Left vs right		
Algorithms	SM $\pm$ SD	
CSP	78.01 $\pm$ 17.01	
GLRCSP	76.62 $\pm$ 15.10	
CCSP1	75.00 $\pm$ 17.08	
CCSP2	77.39 $\pm$ 16.81	
DLCSPPauto	78.32 $\pm$ 16.69	
DLCSPPcv	76.24 $\pm$ 17.04	
DLCSPPdiff	76.24 $\pm$ 17.04	
SSRCSP	77.93 $\pm$ 16.70	
TRCSP	78.78 $\pm$ 15.59	
WTRCSP	78.47 $\pm$ 15.66	
SRCSP	77.70 $\pm$ 16.86	
I-BCI	84.41 $\pm$ 11.88	

(b) Left vs foot		
Algorithms	SM $\pm$ SD	
CSP	81.64 $\pm$ 16.23	
GLRCSP	82.25 $\pm$ 13.77	
CCSP1	81.64 $\pm$ 15.50	
CCSP2	81.25 $\pm$ 15.65	
DLCSPPauto	81.71 $\pm$ 16.04	
DLCSPPcv	80.32 $\pm$ 17.12	
DLCSPPdiff	81.17 $\pm$ 15.06	
SSRCSP	83.18 $\pm$ 13.53	
TRCSP	80.09 $\pm$ 17.31	
WTRCSP	83.26 $\pm$ 15.12	
SRCSP	83.80 $\pm$ 13.83	
I-BCI	90.05 $\pm$ 10.29	

(c) Left vs tongue		
Algorithms	SM $\pm$ SD	
CSP	85.80 $\pm$ 12.54	
GLRCSP	84.95 $\pm$ 14.51	
CCSP1	85.03 $\pm$ 14.63	
CCSP2	86.27 $\pm$ 12.68	
DLCSPPauto	85.96 $\pm$ 12.80	
DLCSPPcv	85.34 $\pm$ 13.40	
DLCSPPdiff	80.32 $\pm$ 15.99	
SSRCSP	85.65 $\pm$ 12.47	
TRCSP	85.34 $\pm$ 13.59	
WTRCSP	85.42 $\pm$ 13.32	
SRCSP	84.03 $\pm$ 15.51	
I-BCI	90.97 $\pm$ 10.25	

(d) Right vs foot		
Algorithms	SM $\pm$ SD	
CSP	84.10 $\pm$ 13.23	
GLRCSP	82.79 $\pm$ 14.40	
CCSP1	83.33 $\pm$ 14.87	
CCSP2	82.87 $\pm$ 13.98	
DLCSPPauto	84.26 $\pm$ 13.08	
DLCSPPcv	83.72 $\pm$ 13.21	
DLCSPPdiff	79.24 $\pm$ 16.96	
SSRCSP	83.26 $\pm$ 14.25	
TRCSP	83.26 $\pm$ 12.25	
WTRCSP	84.41 $\pm$ 12.33	
SRCSP	84.41 $\pm$ 11.59	
I-BCI	90.12 $\pm$ 10.09	

(e) Right vs tongue		
Algorithms	SM $\pm$ SD	
CSP	83.80 $\pm$ 14.16	
GLRCSP	83.33 $\pm$ 15.37	
CCSP1	83.26 $\pm$ 15.45	
CCSP2	83.80 $\pm$ 14.19	
DLCSPPauto	84.26 $\pm$ 13.73	
DLCSPPcv	82.64 $\pm$ 15.76	
DLCSPPdiff	83.02 $\pm$ 15.78	
SSRCSP	83.26 $\pm$ 15.30	
TRCSP	85.03 $\pm$ 13.56	
WTRCSP	84.49 $\pm$ 13.58	
SRCSP	83.33 $\pm$ 15.28	
I-BCI	88.58 $\pm$ 11.32	

(f) Foot vs tongue		
Algorithms	SM $\pm$ SD	
CSP	72.76 $\pm$ 10.36	
GLRCSP	73.38 $\pm$ 10.51	
CCSP1	72.38 $\pm$ 12.01	
CCSP2	72.45 $\pm$ 10.38	
DLCSPPauto	73.15 $\pm$ 10.32	
DLCSPPcv	71.60 $\pm$ 10.69	
DLCSPPdiff	72.07 $\pm$ 10.07	
SSRCSP	73.46 $\pm$ 9.11	
TRCSP	72.92 $\pm$ 10.31	
WTRCSP	73.61 $\pm$ 9.02	
SRCSP	73.92 $\pm$ 9.37	
I-BCI	83.26 $\pm$ 6.82	

Table 5.3: Classification performance (SM in % and SD) obtained for each subject for the TRCSP-SVM and I-BCI on the left vs right hand task -BCI competition IV dataset-IIa.

Algorithm	Subjects									SM $\pm$ SD
	A1	A2	A3	A4	A5	A6	A7	A8	A9	
TRCSP SVM	90.27	59.72	96.53	71.52	63.19	70.13	82.63	96.53	92.36	80.32 14.46
I-BCI	93.75	71.53	98.61	76.39	69.44	74.31	83.33	97.92	94.44	84.41 ± 11.88

5.1.1.1 Comparison of I-BCI with Non-linear Classifier

The techniques reported in the literature modify the CSP algorithm to handle non-stationarity in EEG data. However, regularizing CSP does not eradicate completely, the effects of non-stationarity. The nature of classifier (linear/non-linear) affects the accuracy, but it is not able to handle the non-stationary data. In order to further explain the point, one of the better performing modifications of CSP, TRCSP is considered and instead of LDA, the non-linear SVM classifier is used and its performance is compared with I-BCI for the left vs right hand classification task in Table 5.3. It may be seen from the table that I-BCI performs better than TRCSP-SVM, even with the use of a non-linear classifier, for all the subjects. It achieves the highest improvement in accuracy for the subject A2. This is due to the ability of I-BCI to handle non-stationarity. It should also be noted that I-BCI achieves better subject-wise standard deviation than TRCSP making it more consistent across the subjects.

5.1.1.2 Statistical Comparison

There are fifty-four data sets in total (nine subjects performing six classification tasks) and twelve algorithms (I-BCI and eleven compared algorithms (CSP, CSSP1, CSSP2, GLRCSP, DLCSP: (DLCSP_{auto}, DLCSP_{diff}, DLCSP_{cv}), SSRCSPP, TRCSP, WTRCSP and SRCSP). Since the proposed algorithm is compared with multiple algorithms over multiple data sets, the Friedman test followed by the Bonferroni-Dunn rank test is used to establish the statistical significance of the proposed algorithm [193]. The Friedman test is used to compare multiple algorithms over multiple data sets. It compares the average ranks of the algorithms. In this study, the algorithms are ranked based on their

mean performances across all the data sets using a fractional ranking scheme that gives average rank to the algorithms with equal mean accuracy. The next best performing algorithm gets a rank according to its position. The detailed results for the ranks for all the classification tasks is provided in Table 5.5. The null hypothesis states that all algorithms are equal and hence their average ranks must be equal. If the measured F-score is greater than the F-statistic at 95% confidence level, then one rejects the equality of mean hypothesis. The F-score obtained using a nonparametric Friedman test is 28.42, that is greater than the F-statistic at 95% confidence level ($F_{11,583,0.025}$ is 2.014), i.e., $28.42 > 2.014$. Hence, one can reject the mean equality hypothesis at a confidence level of 95%. Since the Friedman null hypothesis is rejected, a pairwise comparison using a Bonferroni-Dunn rank test is conducted to highlight the performance, significance of I-BCI classifier with respect to other algorithms. The Bonferroni-Dunn rank test states that the performance of an algorithm is statistically significant than the other algorithm if the difference between average ranks is greater than the Critical Difference (CD). The CD value calculated is 0.971 at 95% confidence level. The mean rank of all the algorithms is given in Table 5.4. It may be seen from the Table 5.4 that the difference in mean rank of proposed algorithm and all the other methods is greater than CD at 95% confidence level. Hence, it can be concluded that I-BCI has outperformed all the other algorithms considered in the study with 95% confidence level.

5.1.2 Performance of I-BCI in Handling Intrasection Non-stationarity - BCI Competition III Data Set-IIIa

1. *Description:* The data set-IIIa from the BCI competition III [192] consists of three subjects. The EEG signal is recorded using 60 electrodes. This study aims to classify the left hand (class 1) and right hand (class 2) motor-imagery tasks. The training and testing data set for subject B1 consist of 90 trials each with 45 trials per class. The data set for subjects B2 and B3 consist of 60 trials each with 30 trials per class.
2. *Results:* Table 5.6 shows the performance of I-BCI on the three subjects for which the data is recorded in a single session (intrasection) across multiple runs. Its performance is compared with CSP, GLRCSP, CCSP1, CCSP2, DLCSP, SSRCSF,

Table 5.4: Mean Rank of the compared algorithms (I-BCI, CSP, CCSP1, CCSP2, GLRCSP, DLCSP: (DLCSPauto, DLCSPdiff, DLCSPcv), SSRCSP, TRCSP, WTRCSP and SRCSP) across all the fifty four data sets (nine subjects performing six classification tasks).

Algorithm	Mean Rank	Diff. in Mean Rank w.r.t I-BCI
CSP	6.89	5.70
GLRCSP	7.36	6.17
CCSP1	7.42	6.23
CCSP2	7.05	5.86
DLCSPauto	5.92	4.73
DLCSPcv	8.04	6.85
DLCSPcvdiff	8.73	7.54
SSRCSP	6.48	5.29
TRCSP	6.66	5.47
WTRCSP	6.04	4.85
SRCSP	6.24	5.05
I-BCI	1.19	

TRCSP, WTRCSP and SRCSP. It is clearly visible from this table that I-BCI performs better than all other algorithms on an average. The next best performing algorithm is DLCSPcv whose average performance is 3% less than I-BCI. This shows that I-BCI is able to handle non-stationarity in an intrasession setting.

5.1.3 Discussion on the Synergy of the Hybrid and the Contribution of Individual Components, RoCSP and SRIT2NFIS

In this section, a discussion on the individual components and the synergy of RoCSP and SRIT2NFIS is provided. For this purpose, the performance of CSP-LDA, best performing TRCSP-LDA, RoCSP-LDA, CSP-SRIT2NFIS and I-BCI (RoCSP+SRIT2NFIS) is considered using data set-IIa of BCI competition IV for left-right classification task in Table 5.7. From the Table 5.7, it can be seen that subject A9 does not have artifacts/non-stationarity, hence, the performance of CSP-LDA, RoCSP-LDA, CSP-SRIT2NFIS and I-BCI is almost same. In the case of subject A3, the EEG signal do not have artifacts in the training set leading to the same performance of CSP, RoCSP. However, there is an improvement in SRIT2NFIS due to its ability to handle intersession variability. Next,

Table 5.5: Detailed results for the Bonferroni Dunn rank test.

Tasks	Algorithms(Rank)											
	CSP	GLRCSP	CCSP1	CCSP2	DLCSFauto	DLCSFcv	DLCSFcvdiff	SSRCSP	TRCSP	WTRCSP	SRCSP	I-BCI
leftvsright	88.89(6)	86.11(1.5)	86.11(1.5)	88.89(6)	88.89(6)	88.89(6)	88.89(6)	88.89(6)	88.89(6)	88.89(6)	88.89(6)	93.75(1)
	51.39(9.5)	58.33(3)	60.42(2)	53.47(6.5)	51.39(9.5)	50.69(11.5)	50.69(11.5)	53.47(6.5)	54.17(5)	54.86(4)	52.78(8)	71.53(1)
	96.53(7)	93.75(11.5)	97.22(2.5)	97.22(2.5)	96.53(7)	96.53(7)	96.53(7)	97.22(2.5)	96.53(7)	96.53(7)	96.53(7)	98.61(1)
	70.14(6.5)	67.36(11)	56.94(12)	70.14(6.5)	70.14(6.5)	70.14(6.5)	70.14(6.5)	70.14(6.5)	70.83(2)	70.14(6.5)	70.14(6.5)	76.39(1)
	54.86(10)	55.56(8)	49.31(12)	54.17(11)	56.94(4)	55.56(8)	55.56(8)	56.25(5.5)	62.51(5)	65.97(2)	56.25(5.5)	69.44(1)
	71.53(2.5)	65.28(8.5)	65.28(8.5)	68.06(5)	71.53(2.5)	62.51(10.5)	62.51(10.5)	68.75(4)	67.36(6)	61.81(12)	65.97(7)	74.31(1)
	81.25(7)	81.25(7)	81.25(7)	79.17(11.5)	81.94(2.5)	81.25(7)	81.25(7)	81.25(7)	81.25(7)	81.25(7)	81.94(2.5)	83.33(1)
	93.75(9.5)	93.75(9.5)	93.75(9.5)	95.14(5.5)	93.75(9.5)	93.75(9.5)	93.75(9.5)	97.22(2)	95.83(3.5)	95.83(3.5)	95.14(5.5)	97.92(1)
	93.75(2.5)	88.19(9.5)	88.19(9.5)	90.28(7.5)	93.75(2.5)	86.81(11.5)	86.81(11.5)	90.28(7.5)	91.67(4.5)	90.97(6)	91.67(4.5)	94.44(1)
	97.92(2.5)	93.75(11.5)	96.53(6.5)	96.53(6.5)	97.92(2.5)	93.75(11.5)	96.53(6.5)	97.22(4)	95.14(9)	96.53(9)	94.44(10)	100(1)
leftvsfoot	69.44(9.5)	69.44(9.5)	69.44(9.5)	72.22(3.5)	70.83(5.5)	69.44(9.5)	72.92(2)	70.83(5.5)	69.44(9.5)	69.44(9.5)	72.22(3.5)	81.25(1)
	93.06(7)	93.06(7)	93.06(7)	93.06(7)	93.06(7)	93.06(7)	93.06(7)	93.06(7)	92.36(12)	93.06(7)	93.75(2)	96.53(1)
	75.69(6.5)	75.69(6.5)	75.69(6.5)	71.53(11.5)	75.69(6.5)	75.69(6.5)	75.69(6.5)	74.31(10)	71.53(11.5)	86.11(2.5)	86.11(2.5)	93.75(1)
	61.11(9)	61.81(7)	63.89(6)	68.06(3)	61.11(9)	53.47(12)	61.11(9)	69.44(2)	54.17(11)	66.67(4)	63.19(6)	75.69(1)
	56.94(8)	67.36(2)	56.94(8)	53.47(12)	56.94(8)	56.94(8)	56.94(8)	63.89(3.5)	56.94(8)	56.94(8)	63.89(3.5)	74.31(1)
	98.61(6.5)	98.61(6.5)	98.61(6.5)	98.61(6.5)	98.61(6.5)	98.61(6.5)	98.61(6.5)	98.61(6.5)	98.61(6.5)	98.61(6.5)	98.61(6.5)	100(1)
	86.81(7)	86.81(7)	86.81(7)	84.03(12)	86.81(7)	86.81(7)	86.81(7)	86.81(7)	87.5(2)	86.81(7)	86.81(7)	90.97(1)
	95.14(4)	93.75(11)	93.75(11)	93.75(11)	94.44(8)	95.14(4)	94.44(8)	94.44(8)	95.14(4)	95.14(4)	95.14(4)	97.92(1)
	96.53(9)	97.22(5.5)	58.33(12)	98.61(2.5)	96.53(9)	96.53(9)	96.53(9)	97.92(4)	96.53(9)	96.53(9)	98.61(2.5)	100(1)
	67.36(2)	61.81(8.5)	61.81(8.5)	65.28(5)	66.67(3.5)	62.51(6.5)	59.72(11)	66.67(3.5)	60.42(10)	62.51(6.5)	61.81(8.5)	73.61(1)
leftvsstone	94.44(6)	94.44(6)	94.44(6)	94.44(6)	94.44(6)	94.44(6)	85.42(12)	93.75(11)	94.44(6)	94.44(6)	94.44(6)	97.22(1)
	90.28(5.5)	90.28(5.5)	90.28(5.5)	88.89(9.5)	90.97(2)	90.28(5.5)	80.56(12)	87.5(11)	90.28(5.5)	90.28(5.5)	88.89(9.5)	93.06(1)
	69.44(8)	65.28(10)	69.44(8)	73.61(2)	69.44(8)	70.14(5.5)	70.14(5.5)	72.22(3)	72.22(3)	70.14(5.5)	61.81(11)	79.17(1)
	71.53(6.5)	71.53(6.5)	71.53(6.5)	71.53(6.5)	71.53(6.5)	71.53(6.5)	63.89(12)	71.53(6.5)	71.53(6.5)	72.22(2)	68.06(11)	80.56(1)
	97.22(6.5)	97.22(6.5)	97.22(6.5)	97.22(6.5)	97.22(6.5)	97.22(6.5)	97.22(6.5)	97.22(6.5)	97.22(6.5)	97.22(6.5)	97.22(6.5)	98.61(1)
	90.28(7.5)	90.28(7.5)	90.28(7.5)	90.28(7.5)	90.97(2)	90.28(7.5)	90.28(7.5)	90.28(7.5)	90.28(7.5)	90.28(7.5)	90.28(7.5)	97.22(1)
	95.14(9)	96.53(3)	96.53(3)	96.53(3)	95.83(6)	95.14(9)	95.14(9)	95.14(9)	95.14(9)	95.14(9)	95.14(9)	99.31(1)
	97.22(11)	97.22(11)	97.22(11)	98.61(7)	99.31(4)	97.22(11)	100(1.5)	99.31(4)	98.61(7)	98.61(7)	98.61(7)	100(1.5)
	79.86(5.5)	68.75(11)	79.86(5.5)	75(10)	79.86(5.5)	79.86(5.5)	54.17(12)	78.47(8)	80.56(2.5)	80.56(2.5)	80.56(2.5)	86.11(1)
	91.67(7)	91.67(7)	91.67(7)	91.67(7)	91.67(7)	91.67(7)	91.67(7)	91.67(7)	91.67(7)	91.67(7)	91.67(7)	97.92(1)
rightvsfoot	88.19(6)	88.19(6)	88.19(6)	88.19(6)	88.19(6)	88.19(6)	88.19(6)	88.19(6)	84.03(11.5)	84.03(11.5)	84.03(11.5)	92.36(1)
	64.58(4.5)	61.81(8)	55.56(11)	59.72(9)	64.58(4.5)	63.19(7)	54.86(12)	57.64(10)	65.97(3)	63.89(6)	67.36(2)	75(1)
	63.19(9.5)	63.19(9.5)	63.19(9.5)	63.19(9.5)	63.89(5.5)	63.19(9.5)	63.19(9.5)	64.58(4)	63.89(5.5)	65.28(2.5)	65.28(2.5)	72.92(1)
	99.31(3.5)	99.31(3.5)	99.31(3.5)	96.53(7)	99.31(3.5)	99.31(3.5)	90.28(12)	95.83(8)	95.14(10)	95.14(10)	95.14(10)	99.31(3.5)
	88.89(7)	88.89(7)	88.89(7)	88.89(7)	88.89(7)	88.89(7)	88.89(7)	88.89(7)	88.89(7)	88.89(7)	88.89(7)	93.75(1)
	81.94(11)	81.94(11)	86.11(5)	84.03(7.5)	82.64(9)	81.94(11)	84.03(7.5)	86.11(5)	81.94(11)	87.5(3)	88.19(2)	93.75(1)
	100(6.5)	100(6.5)	100(6.5)	100(6.5)	100(6.5)	100(6.5)	100(6.5)	100(6.5)	100(6.5)	100(6.5)	100(6.5)	100(6.5)
	63.89(9.5)	63.89(9.5)	63.89(9.5)	69.44(2.5)	65.28(6)	63.89(9.5)	63.89(9.5)	68.75(4)	73.61(4)	69.44(2.5)	66.67(5)	73.61(1)
	96.53(11)	97.92(7)	97.92(7)	97.92(7)	96.53(11)	96.53(11)	98.61(4.5)	99.31(3)	100(1.5)	99.31(3)	97.92(7)	100(1.5)
	84.72(9)	84.72(9)	84.72(9)	84.72(9)	85.42(4)	84.72(9)	86.11(2)	85.42(4)	85.42(4)	85.42(4)	85.42(4)	90.97(1)
rightvsstone	65.97(4)	61.81(9)	62.51(7)	65.97(4)	66.67(3)	65.97(4)	61.81(9)	71.53(2)	71.53(2)	65.28(5)	60.42(12)	74.31(1)
	68.75(3.5)	66.67(5.5)	65.28(9)	65.28(9)	69.44(2)	63.19(12)	63.89(11)	66.67(5.5)	65.28(9)	68.75(3.5)	74.31(1)	74.31(1)
	97.22(6)	97.92(2)	97.92(2)	96.53(10)	97.22(6)	97.22(6)	97.22(6)	97.22(6)	97.22(6)	95.83(12)	96.53(10)	97.92(1)
	90.97(7)	90.97(7)	90.97(7)	90.97(7)	90.97(7)	90.97(7)	90.97(7)	90.97(7)	92.36(1)	90.97(7)	92.36(1)	93.75(1)
	86.11(7)	86.11(7)	86.11(7)	86.11(7)	86.11(7)	86.11(7)	86.11(7)	85.42(11.5)	85.42(11.5)	86.81(2.5)	90.97(7)	90.97(1)
	72.22(7.5)	72.22(7.5)	72.22(7.5)	72.22(7.5)	72.22(7.5)	72.22(7.5)	72.22(7.5)	74.31(2)	73.61(1)	72.92(3.5)	72.22(7.5)	77.78(1)
	70.83(8.5)	71.53(6.5)	71.53(6.5)	72.22(5)	70.83(8.5)	71.53(12)	72.92(11)	75.69(2)	68.06(10)	72.92(3.5)	72.92(3.5)	82.64(1)
	68.75(7)	68.75(7)	68.75(7)	68.75(7)	68.75(7)	68.75(7)	68.75(7)	68.75(7)	68.75(7)	68.75(7)	68.75(7)	79.86(1)
	55.56(9.5)	55.56(9.5)	55.56(9.5)	58.03(3.5)	56.25(5.5)	55.56(9.5)	55.56(9.5)	56.25(5.5)	56.25(5.5)	59.03(3.5)	59.72(2)	76.39(1)
	67.36(8)	67.36(8)	56.94(12)	59.72(11)	67.36(8)	67.36(8)	67.36(8)	69.44(3)	70.14(2)	68.75(4.5)	68.75(4.5)	78.47(1)
footvsstone	66.67(10)	68.06(4)	68.06(4)	65.97(12)	67.36(7)	66.67(10)	66.67(10)	69.44(2)	68.06(4)	67.36(7)	67.36(7)	79.86(1)
	81.94(7)	81.94(7)	81.94(7)	81.94(7)	81.94(7)	81.94(7)	81.94(7)	81.94(7)	81.94(7)	81.94(7)	81.94(7)	86.81(1)
	80.56(9.5)	86.11(2.5)	82.64(6)	82.64(6)	91.67(2)	80.56(9.5)	79.17(11.5)	79.17(11.5)	84.72(5)	81.25(7.5)	85.42(4)	90.97(1)
	90.97(3)	88.89(8.5)	90.28(4)	88.89(8.5)	91.67(2)	89.58(5.5)	89.58(5.5)	86.81(12)	88.89(8.5)	88.89(8.5)	88.89(8.5)	96.53(1)
	81.02(6.89)	80.56(7.36)	80.11(7.42)	80.67(7.05)	81.28(5.92)	78.68(8.04)	78.68(8.73)	81.12(6.48)	80.90(6.56)	81.61(6.04)	81.20(6.24)	87.90(1.19)

Table 5.6: Classification performance (SM in % and SD) obtained for each subject for the standard CSP and its variants on the intrasession BCI competition III data set-IIIa.

Algorithm	Subjects			SM $\pm$ SD
	B1	B2	B3	
CSP	95.56	61.67	93.33	83.52 $\pm$ (18.95)
GLRCSP	95.56	61.67	90	81.41 $\pm$ (18.17)
CCSP1	98.89	45	93.33	79.07 $\pm$ (29.63)
CCSP2	95.56	61.67	93.33	83.52 $\pm$ (18.95)
DLCSP _{auto}	94.44	63.33	95	84.25 $\pm$ (18.12)
DLCSP _{cv}	95.56	78.33	93.33	89.07 $\pm$ (9.37)
DLCSP _{cvdiff}	95.56	66.67	93.33	85.18 $\pm$ (16.07)
SSRCSP	95.56	61.67	96.67	84.63 $\pm$ (19.89)
TRCSP	98.89	56.67	93.33	82.96 $\pm$ (22.94)
WTRCSP	98.89	71.67	93.33	87.96 $\pm$ (14.38)
SRCSP	96.67	53.33	93.33	81.11 $\pm$ (24.11)
I-BCI	100	78.83	100	92.94 $\pm$ (12.22)

Table 5.7: Classification performance (SM in % and SD) obtained for each subject for the CSP-LDA, TRCSP-LDA, RoCSP-LDA, CSP-SRIT2NFIS and I-BCI on the left vs right hand task -BCI competition IV dataset-IIa.

Subjects	Algorithms				
	CSP-LDA	RoCSP-LDA	TRCSP-LDA	CSP-SRIT2NFIS	I-BCI
A1	88.89	88.89	88.89	91.67	93.75
A2	51.39	61.11	54.17	63.89	71.53
A3	96.53	96.53	96.53	97.22	98.61
A4	70.14	70.14	70.83	72.92	76.39
A5	54.86	61.81	62.50	63.89	69.44
A6	71.53	71.53	67.36	72.22	74.31
A7	81.25	81.25	81.25	82.64	83.33
A8	93.75	95.83	95.83	96.53	97.92
A9	93.75	93.75	91.67	93.75	94.44
SM	78.01	80.09	78.78	81.63	84.41
SD	17.01	14.36	15.59	13.72	11.88

the case of subject A2 is highlighted (shown in bold), which is affected by both artifacts and non-stationarity leading to a lower performance of the CSP algorithm (51.39%). In this case, the RoCSP algorithm performs better than the CSP algorithm by 10%. This is due to the ability of the RoCSP algorithm to eliminate those trials affected by artifacts resulting in a robust projection matrix and better spatial filters. However, RoCSP

alone cannot handle non-stationarity. Here, CSP-SRIT2NFIS is able to handle the non-stationarity and provide an improved performance of 12% over CSP but it suffers from poor quality of the CSP projection matrix. The synergy of both the methods, I-BCI performs better than both RoCSP and CSP-SRIT2NFIS and provides an increment of 20 %. This highlights the importance of the synergy in cases where the data is both non-stationary and also affected by artifacts. From the Table 5.7, it can also be seen that I-BCI handles non-stationarity/artifacts better than the next best performing TRCSP for this task. It should also be noted that I-BCI achieves the lowest standard-deviation across subjects among all the compared algorithms, thereby highlighting its consistency in performance across subjects.

5.2 Summary

In this chapter, an intelligent-brain computer interface system is presented to handle the non-stationarity present in EEG data. I-BCI uses the non-stationary features generated by the RoCSP algorithm and models them as uncertainty in the data distribution space using the SRIT2NFIS algorithm. The performance of the I-BCI system in handling the non-stationarity in EEG signals generated by motor-imagery tasks in intersession/intrasession settings has been highlighted using two publicly available BCI competition data sets (Dataset IIIa from BCI Competition III and BCI Competition data set IV-IIa). The results have been compared with other existing algorithms in the literature. Results show that I-BCI outperformed the compared algorithms in simple tasks by an average of 3-5% and in complex tasks by 8-10%. These results also show that I-BCI improvements were even more significant for subjects where the initial CSP performance was lower. A Friedman test followed by post-hoc Bonferroni-Dunn rank test proved that the performance improvement of I-BCI over other algorithms is statistically significant with a 95% confidence level.

The I-BCI system employs a common frequency range for all the subjects, however, there is a variation across frequency responses of different subjects. In the next chapter, a mechanism is developed that finds the optimal subject-specific spatio-spectral filters.

Chapter 6

A Discriminative Subject-Specific Spatio-Spectral Filter Selection Approach for EEG based Motor-Imagery Task Classification

In the previous chapter, an Intelligent-Brain Computer Interface (I-BCI) system has been proposed that addresses the key problems of non-stationarity and influence of artifacts in EEG based BCI systems. In this chapter, the problem of finding the optimal subject-specific spatio-spectral filters for motor-imagery based BCI is addressed. During mental imagination of different motor-imagery tasks, event related de-synchronization/ synchronization patterns are generated in the EEG. Here, Event-Related De-synchronization (ERD) pattern (short-lasting attenuation or blocking of rhythms) occurs in the (μ (8-12) Hz and β (14-30) Hz) frequency bands. The opposite phenomenon of ERD is called Event-Related Synchronization (ERS) pattern, where an increase in spectral peak occurs in μ and β bands. These patterns are useful phenomena in motor-imagery tasks for accurate detection of the task. For more details on ERD, and ERS, refer to [16, 30]. These ERD/ERS patterns occur in different frequency ranges for different subjects [92]. Based on these subject-specific patterns, different spatial locations of the brain have different activations which makes feature extraction from a specific location challenging. Hence, there is a need to find the subject-specific frequency ranges (using band-pass filters referred to as spectral filters) followed by spatial filters for accurate motor-imagery task classification. Together, this process is referred to as spatio-spectral filter selection.

The accurate detection of spatio-spectral filters for a given subject is challenging due to the presence of non-stationarity in EEG signals [23].

The research works in the literature for solving the subject-specific spatio-spectral filter selection can be grouped into three major categories based on the nature of EEG signal processing. In the first category, time-delayed signals are embedded into raw EEG signals and the Common Spatial Pattern (CSP) algorithm [17, 18] is used to optimize both the spatial and spectral finite impulse response filters [49, 93]. In the Common Spatio-Spectral Pattern (CSSP) method [49], the CSP algorithm has been extended by appending a time-delayed EEG signal as the signal from additional channels resulting in additional finite impulse response filters at each electrode. However, the CSSP algorithm is not flexible due to the use of a single time-delay filter [93]. Later, [93] extended the CSSP algorithm by allowing a larger time-delay. However, the selection of time-delay parameter is computationally expensive [94]. In the second category, the spectral filters are optimized in the frequency domain [94, 95]. Here, an iterative procedure is employed in which the spatial filters are optimized using the CSP algorithm followed by the optimization of the spectral weighting coefficients in the frequency domain. However, the optimization in the frequency domain is difficult as ERD/ERS patterns occur in specific frequency bands which makes the tuning difficult. In the last category, the frequency range is discretized into multiple smaller bands using a filter bank. Here, researchers have focused on finding the subject-specific spatio-spectral filters by constructing a filter bank and applying the CSP algorithm on each of the frequency bands. In [96, 97], a broad frequency range was divided into sub-bands and a reduced set of filters were selected. In [96], a score for each sub-band was calculated using the Linear Discriminant Analysis (LDA) and the top bands with higher scores were selected. In the Discriminative Filter Bank CSP (DFBCSP) [97], the Fisher ratio of signals at each band output was used to select the optimum bands. These algorithms only find the optimal spectral filters that look at the frequencies and they do not consider the spatial aspects. In order to address the spatial aspects, several algorithms were proposed with an aim to optimize both the spectral and spatial filters [92, 98–100]. These algorithms employed a mutual information based approach to find the spatio-spectral filters. Filter Bank CSP (FBCSP) [92] employed a mutual information based filter selection approach to select

a subset of filters generated by a filter bank, followed by the band selection. In the Optimal Spatio-Spectral Filter Network (OSSFN) work [98], spatial filter optimization was performed at frequency band levels followed by the selection of that band with the maximal mutual information, while in [99], the spatio-spectral filters were optimized simultaneously. All these algorithms used a fixed filter bank. Recently, Bayesian Spatio-Spectral Filter Optimization (BSSFO) [100] was proposed that assigned different weights to different frequency bands. It employed a wrapper around method based on Bayesian framework for optimizing the spatio-spectral filters. It also combined the outputs from multiple classifiers using the weights of different frequency bands. The use of multiple classifiers makes this approach complex in nature. It should be pointed out here that all the above methods use the CSP algorithm as the base to generate the features for the classification of motor-imagery tasks. However, the features generated using the CSP algorithm are inherently non-stationary in nature [23, 71]. The earlier works either employ LDA or Support Vector Machine (SVM) as the classifier that are unable to handle non-stationary EEG signals [147] effectively and using a CSP based approach with these classifiers might lead to a lower classification performance. This is the main motivation behind this work that develops a new approach that is capable of finding the optimal subject-specific spatio-spectral filters in the presence of non-stationarity in EEG signals and also provide good classification performance.

This chapter presents such an approach using the SRIT2NFIS classifier to find the subject-specific spatio-spectral filters for motor-imagery based BCI. SRIT2NFIS employs an interval type-2 fuzzy system capable of handling noisy and non-stationary EEG signals effectively. The spatio-spectral filter selection is cast into the problem of band selection in a filter bank followed by spatial filter selection in the selected bands. It has a two-step approach, where in the first step spectral filter selection is performed, followed by the second step where the spatial filter selection is done. This approach is shown in Fig. 6.1. In the first step, SRIT2NFIS is used to eliminate unnecessary bands from a filter bank to arrive at a set of optimal spectral filters, referred to as Spectral-I-BCI (S-IBCI) method. Here, frequency bands are iteratively eliminated such that the overall classification performance is better. In this step, the raw EEG signals are band-passed using six equally partitioned bands of 4 *Hz* each as given in [92, 97], from 8 to 32

Figure 6.1: Block diagram of two-step spatio-spectral filtering using SRIT2NFIS classifier.

Hz covering the mu and beta rhythms followed by the RoCSP algorithm based feature extraction for each of the bands. The non-stationary nature of features generated by the RoCSP algorithm is handled as uncertainty using the SRIT2NFIS classifier. In the second step, SRIT2NFIS takes those RoCSP filters from the selected bands in the first step and eliminates unnecessary spatial filters to obtain an optimal set of spatial filters. A similar approach has been employed to remove the spatial filters from the selected spectral bands. Here, the spatial filters corresponding to the bands selected in first step are iteratively eliminated such that the overall classification performance is better. At the end of the second step, the proposed method is able to identify the desired spatio-spectral filters for a given subject. This procedure is referred to as Spatio-Spectral-I-BCI (SS-IBCI) method.

Performances of S-IBCI and SS-IBCI are evaluated using two data sets viz. the data set-IIa from the BCI competition IV [116] (six classification tasks) and the data set-IVa [192] from the BCI competition III [18]. The evaluation has been carried out as described below: First, the performance of the S-IBCI algorithm is evaluated in selecting the spectral filters for the classification tasks, followed by the evaluation of SS-IBCI algorithm in the selection of spatio-spectral filters. The performance of S-IBCI and SS-IBCI is compared with other existing algorithms like CSSP, FBCSP, DFBCSP, OSSFN (OSSFNwFBCSP-OSSFN utilizing FBCSP, OSSFNwDFBCSP-OSSFN utilizing DFBCSP) and BSSFO.

The rest of the chapter is organized as follows. In Section 6.1, the spatio-spectral feature selection using the SRIT2NFIS algorithm is presented. Section 6.2 presents the performance evaluation of the proposed approach and Section 6.3 presents the summary of the chapter.

6.1 Spatio-Spectral Filter Selection using SRIT2NFIS

Let the total number of bands be given by B . The objective of S-IBCI and SS-IBCI is to select a subset of bands B_{new} and corresponding spatial filters such that the classification accuracy is maximized. The problem is tractable since there are a constant number of bands. A two-step approach is employed to find the subject-specific spatio-spectral filters

using the SRIT2NFIS classifier. The idea is to use the SRIT2NFIS algorithm to find the desired spatio-spectral filters in a two-step approach. In the first step, the spectral filters are found by eliminating those bands from a filter bank that do not affect the SRIT2NFIS classification accuracy. In the second step, SRIT2NFIS is used to find the optimal spatial filters by eliminating the irrelevant spatial filters obtained from those frequency bands that were selected in the first step. First, the generation of features using a filter bank is described. Next, the SRIT2NFIS classifier for motor-imagery task classification is highlighted followed by the selection procedure for both spectral and spatial filters.

6.1.1 Feature Generation from EEG using a Filter Bank

The filter bank comprises of six frequency bands of 4 *Hz* each [92] in the range of 8-32 *Hz* (8-12 *Hz*, 12-16 *Hz* and so on) covering the mu and beta rhythms. It has been shown in [97, 194], that the frequency bands of width 4 *Hz* yield a stable frequency response and provide the best results in comparison to other bands of varying widths in the range of 2-6 *Hz*. The EEG signals generated from the motor-imagery tasks are processed using each of these frequency bands. The band passed signals are then spatially filtered using the RoCSP algorithm (explained in detail in Chapter 3) to generate the log-variance features for each frequency band [17, 50].

The log-variance feature computation using the RoCSP algorithm for a two-class problem is summarized below. It aims to improve the classification performance of the CSP algorithm [17] by eliminating redundant channels in the computation of the log-variance features. It eliminates the noisy trials in each channel followed by an effect-size based channel selection. RoCSP computes the projection matrix $\mathbf{V} = (\mathbf{U}^T \mathbf{P})$ using the selected trials and channels. The projection matrix $\mathbf{V}$ are called spatial filters. From $\mathbf{V}$, the three most dominant pair of spatial filters $\mathbf{V}' = [\mathbf{v}_1 \ \mathbf{v}_2 \ \mathbf{v}_3 \ \mathbf{v}_{2m-2} \ \mathbf{v}_{2m-1} \ \mathbf{v}_{2m}]$ for motor-imagery task 1 and task 2 are chosen, respectively. The projected signal from the selected spatial filters can be used to map a trial $\mathbf{E}$ is given by:

$$\mathbf{Z} = \mathbf{V}' \times \mathbf{E} \quad (6.1)$$

In this chapter, the log-variances of the signals are computed as features and are given by:

$$\mathbf{b}_j = \log(\text{var}(\mathbf{Z})); \ j = 1, 2, \dots, 6 \quad (6.2)$$

where, $\mathbf{b}_j = [b_{j1}, \dots, b_{j6}]^T \in \mathbb{R}^6$ is a 6-dimensional feature vector in the j -th band. These spatial features are the log-variances of the projected EEG signal from the selected spatial filters. The spatial filters provide a linear combination of raw EEG signals measured at different locations of the brain. These filters are computed such that they project the EEG signals into an orthogonal space. Equations (6.1) and (6.2) are repeated for every trial to compute the log-variance features.

6.1.2 SRIT2NFIS Motor-imagery Task Classifier

In this section, the description of the SRIT2NFIS architecture and its learning algorithm is summarized. Let the data generated by the j -th band be $\{(\mathbf{b}_j^1, c^1), \dots, (\mathbf{b}_j^t, c^t), \dots, (\mathbf{b}_j^N, c^N)\}$, where, $\mathbf{b}_j^t = [b_{j1}^t, \dots, b_{j6}^t]^T \in \mathbb{R}^6$ is the 6-dimensional input vector of the t^{th} sample in the j -th band and $c^t \in [1, 2]$ is the corresponding class label. The binary class label y^t is given as:

$$y^t = \begin{cases} 1, & \text{If } c^t == 1 \\ -1, & \text{If } c^t == 2. \end{cases} \quad (6.3)$$

The objective of the SRIT2NFIS classifier is to approximate the underlying functional relationship $(\mathbf{b}_j^t \rightarrow \mathbf{y}^t)$ between the input $(\mathbf{b}_j^t)$ and output $(\mathbf{y}^t)$.

6.1.2.1 Self-Regulated Interval Type-2 Neuro-Fuzzy Inference System Learning Architecture

The architecture of SRIT2NFIS (Fig. 6.2) is a five-layered Interval type-2 neuro-fuzzy inference system used to predict the class label for the current sample. Let us assume that the network has grown K rules after processing $t - 1$ data samples. The detailed description of each layer when presenting the t^{th} sample in j -th band is as follows:

Layer 1- Input layer: The input layer consists of six nodes, one each for the six frequency bands. The output of the i^{th} node of the j -th band of the input layer is given by:

$$u_{ji}^t = b_{ji}^t; \quad j = 1, 2, \dots, 6; \quad i = 1, 2, \dots, 6. \quad (6.4)$$

The band dependence j is dropped hereafter for convenience.

Layer 2- Fuzzification layer: Each node in this layer handles the non-stationary nature of the input features as an uncertainty bound by employing an Interval type-2 fuzzy

Figure 6.2: Architecture and learning scheme of SRIT2NFIS.

variable. The membership of the i^{th} feature with the k^{th} rule is given by:

$$\phi_{ik} = \exp \left(-\frac{(u_i^t - \mu_{ik})^2}{2\sigma_k^2} \right) \equiv \phi(\mu_{ik}, \sigma_k, u_i^t), \quad (6.5)$$

where, $\mu_{ik} \in [\mu_{ik}^1, \mu_{ik}^2]$ are the left and right limits of the center and σ_k is the width of k^{th} rule. The footprint of the uncertainty of this membership function is represented as a bounded interval in terms of an upper membership function ϕ^{up} and a lower membership function ϕ^{lo} , given by:

$$\phi_{ik}^{up,t} = \begin{cases} \phi(\mu_{ik}^1, \sigma_k, u_i^t) & u_i^t < \mu_{ik}^1 \\ 1 & \mu_{ik}^1 \leq u_i^t \leq \mu_{ik}^2 \\ \phi(\mu_{ik}^2, \sigma_k, u_i^t) & u_i^t > \mu_{ik}^2 \end{cases} \quad (6.6)$$

$$\phi_{ik}^{lo,t} = \begin{cases} \phi(\mu_{ik}^2, \sigma_k, u_i^t) & u_i^t \leq \frac{(\mu_{ik}^1 + \mu_{ik}^2)}{2} \\ \phi(\mu_{ik}^1, \sigma_k, u_i^t) & u_i^t > \frac{(\mu_{ik}^1 + \mu_{ik}^2)}{2} \end{cases} \quad (6.7)$$

The output of each node in the fuzzification layer is represented by the interval $\Phi_{ik}^t = [\phi_{ik}^{lo,t}, \phi_{ik}^{up,t}]$.

Layer 3- Firing layer: The nodes in this layer represent the upper $F_k^{up,t}$ and lower $F_k^{lo,t}$ membership of a rule given by:

$$F_k^t = [F_k^{lo,t}, F_k^{up,t}]; \quad k = 1, \dots, K \quad (6.8)$$

where

$$F_k^{lo,t} = \prod_{i=1}^6 \phi_{ik}^{lo,t} \quad \text{and} \quad F_k^{up,t} = \prod_{i=1}^6 \phi_{ik}^{up,t}; \quad k = 1, \dots, K. \quad (6.9)$$

Layer 4- Type reduction layer: Each node in this layer converts the Interval type-1 fuzzy set to a fuzzy number for a particular rule using the Nie-Tan type-reduction approach [28]. The output of each node F_k^t is given by:

$$F_k^t = \alpha F_k^{lo,t} + (1 - \alpha) F_k^{up,t}; \quad k = 1, \dots, K \quad (6.10)$$

where, α is the weighted measure of uncertainty. In this study, α is chosen as 0.5.

Layer 5-Output layer: This layer calculates the predicted output of the network given by:

$$\hat{y}^t = \frac{\sum_{k=1}^K w_k F_k^t}{\sum_{p=1}^K F_p^t} \quad (6.11)$$

where, w_k is the output weight connecting the k -th rule with the output node. The predicted class label corresponding to $\hat{y}^t$ is given by $\hat{c}^t$, and is defined as:

$$\hat{c}^t = \begin{cases} 1, & \text{If } \hat{y}^t \geq 0 \\ 2, & \text{otherwise.} \end{cases} \quad (6.12)$$

Next, the self-regulating learning algorithm used in the above fuzzy inference system is presented. The objective of this learning algorithm is to find the rules and the associated parameters to approximate the functional relationship between the inputs and the outputs.

6.1.2.2 Self-Regulated Interval Type-2 Neuro-Fuzzy Inference System Learning Algorithm

The SRIT2NFIS learning algorithm employs the Regularized Projection Based Learning (RPBL) algorithm to calculate the weights of the network. The RPBL algorithm finds the optimal weights of the network by minimizing the total energy of the error function. It is explained in Section 4.3.2 in detail. When a new sample $(\mathbf{b}^t, c^t)$ is presented to the network, the SRIT2NFIS learning algorithm monitors the knowledge content of the sample using the prediction error E^t [149], the spherical potential ψ^t [151] and the maximum rule contribution with respect to the knowledge already present in the network. It then decides whether the sample can be used for a parameter update or an addition of a rule or it can be discarded without learning or it can be considered as a reserve sample for learning at later point of time. For further details, refer to Section 4.3.2.

6.1.3 Spectral Filters Selection using Self-Regulated Interval Type-2 Neuro-Fuzzy Inference System

In this section, the approach to find the subject-specific spectral filters using the SRIT2NFIS algorithm is explained in detail. The SRIT2NFIS algorithm is employed to remove all those bands from the filter bank, whose elimination will not have any effect on the performance. It is referred to as Spectral-I-BCI (S-IBCI). This mechanism is illustrated in Fig. 6.3. The features obtained from the filter bank are passed as input to the SRIT2NFIS algorithm. In each iteration, one of the bands is dropped and the performance is re-computed. If the performance does not drop, the particular band is termed as redundant

and excluded from the final set of bands. This procedure is carried out for all the bands and a final smaller set of bands, $\mathbf{B}_{\text{new}}$ is obtained. This mechanism provides a trade-off between the time taken and classification accuracy. The above procedure is summarized in the Pseudo-code 1.

```

Initialization:
Band set  $\mathbf{B} = \{\mathbf{B}_1 \cdots \mathbf{B}_j \cdots \mathbf{B}_6\}$ 
 $BestTrainAccuracy \leftarrow 0$ 
 $TrainAccuracy \leftarrow 0$ 
Calculate training error using features from all the bands
 $BestTrainAccuracy \leftarrow SRIT2NFIS(\mathbf{B})$ 
for  $j \rightarrow 1$  to 6 do
     $\mathbf{B}_{\text{new}} \leftarrow \mathbf{B} \setminus \mathbf{B}_j$ 
     $TrainAccuracy \leftarrow SRIT2NFIS(\mathbf{B}_{\text{new}})$ 
    if  $TrainAccuracy \geq BestTrainAccuracy$  then
         $BestTrainAccuracy \leftarrow TrainAccuracy.$ 
    else
        concatenate( $\mathbf{B}_{\text{new}}, \mathbf{B}_j$ )
end
Output:  $\mathbf{B}_{\text{new}}$ 

```

Algorithm 1: Pseudo-code for spectral filters selection using SRIT2NFIS.

6.1.4 Spatial Filters Selection using Self-Regulated Interval Type-2 Neuro-Fuzzy Inference System

In this section, the procedure for finding the subject-specific spatial filters is explained in detail. Let the bands selected in the previous step be given as $\mathbf{B}_i$, $\mathbf{B}_j$ and $\mathbf{B}_k$. The corresponding spatial filter set of these bands is given as the filter set, $\mathbf{f} = \{(\mathbf{v}_{i1} \cdots \mathbf{v}_{i2m}) \cdots (\mathbf{v}_{j1} \cdots \mathbf{v}_{j2m}) \cdots (\mathbf{v}_{k1} \cdots \mathbf{v}_{k2m})\}$. Here, the appropriate spatial filters are selected inside the obtained bands after the band elimination procedure. The iterative procedure for finding the spatial filters is shown in Pseudo-code 2 and illustrated in Fig. 6.3. At each iteration, a spatial filter is dropped from the spatial filter set, $\mathbf{f}$ obtained from the selected bands in step one. If the performance of SRIT2NFIS does not drop, this spatial filter is considered redundant and excluded from the spatial filter set. This operation is carried out for all the spatial filters in the spatial filter set and a final set of spatial filters are obtained.

Figure 6.3: Schematic diagram of spatio-spectral filtering using SRIT2NFIS classifier.

The mechanism to find spectral filters followed by spatial filters using SRIT2NFIS is referred to as Spatio-Spectral-I-BCI (SS-IBCI). In the next section, the performance of S-IBCI and SS-IBCI is evaluated using the benchmark data sets.

```

Initialization:
  Filter set  $\mathbf{f} = \{(\mathbf{v}_{i1} \dots \mathbf{v}_{i2m}) \dots (\mathbf{v}_{j1} \dots \mathbf{v}_{j2m}) \dots (\mathbf{v}_{k1} \dots \mathbf{v}_{k2m})\}$ 
   $n \leftarrow |\mathbf{f}|$ 
  BestTrainAccuracy  $\leftarrow 0$ 
  TrainAccuracy  $\leftarrow 0$ 
  Calculate training error using all the spatial filters
  BestTrainAccuracy  $\leftarrow \text{SRIT2NFIS}(\mathbf{f})$ 
  for  $j \rightarrow 1$  to  $n$  do
     $\mathbf{f}_{final} \leftarrow \mathbf{f} \setminus \mathbf{f}_j$ 
    TrainAccuracy  $\leftarrow \text{SRIT2NFIS}(\mathbf{f}_{final})$ 
    if TrainAccuracy  $\geq$  BestTrainAccuracy then
      BestTrainAccuracy  $\leftarrow$  TrainAccuracy.
    else
      concatenate( $\mathbf{f}_{final}, \mathbf{f}_j$ )
  end
  Output:  $\mathbf{f}_{final}$ 

```

Algorithm 2: Pseudo-code for spatial filters selection using SRIT2NFIS.

6.2 Experimental Results

The performances of S-IBCI and SS-IBCI have been evaluated using two publicly available BCI competition data sets, viz., data set-IIa from the BCI competition IV [116] and data set-IVa from the BCI competition III [18]. First, the performance of selecting the desired spectral and spatio-spectral filters is evaluated using nine subjects from the data set-IIa from the BCI competition IV, performing the following six binary classification tasks, viz. left vs right hand, left hand vs foot, left hand vs tongue, right hand vs foot, right hand vs tongue and foot vs tongue. This study is also supported by performance evaluation using additional five subjects from the data set-IVa of BCI competition III.

Performance of S-IBCI and SS-IBCI is compared with the other existing methods like the Common Spatio-Spectral Pattern (CSSP) method [49], the Filter Bank CSP (FBCSP) method [92], the Discriminative Filter Bank CSP (DFBCSP) method [97], the Optimal Spatio-Spectral Filter Network (OSSFN) method [98] (OSSFNwFBCSP-OSSFN

utilizing FBCSP, OSSFNwFBCSP-OSSFN utilizing DFBCSP) and finally the Bayesian Spatio-Spectral Filter Optimization (BSSFO) method [100]. It may be noted here that all these algorithms use the SVM as classifier.

6.2.1 Performance Evaluation of S-IBCI and SS-IBCI using the BCI Competition IV Data Set-IIa

Data set IIa [116] consists of data of nine subjects. The EEG signals are recorded using twenty-two non-invasive electrodes and sampled at two hundred fifty *Hz*. The experiment is conducted over two sessions on two different days for each subject. Each session consists of two hundred eighty-eight (seventy-two for each class) trials which are distributed over six runs separated by short breaks. In the experiment, subjects performed the movement of left hand (class one), right hand (class two), foot (class three) and tongue (class four) motor-imagery tasks. The training and testing data consists of seventy-two trials for each class.

Table 6.1 shows the selected bands and classification accuracies of S-IBCI and SS-IBCI for each subject for all the six classification tasks (left vs right hand classification, left vs foot classification, left vs tongue classification, right vs foot classification, right vs tongue classification and foot vs tongue classification). It may be observed from the Table 6.1 that both S-IBCI and SS-IBCI achieve the highest accuracy for left hand vs foot motor-imagery task implying that these motor-imagery tasks are most distinguishable, closely followed by the right vs foot and the left vs tongue tasks. It may also be noted that the spatio-spectral filter selection leads to improvement in the average classification performance for all the classification tasks. Based on the experimental results, existence of common frequency bands of ((8-16) *Hz* and (20-32) *Hz*) for all subjects across the classification tasks were observed. If these common frequency bands across all subjects are used, the performance is comparable with individual spatial-spectral filter selection.

In order to further study the effect of the order of spatial and spectral filtering, three experiments were conducted: Case A) Simultaneous spatial and spectral filtering. Case B) Spatial filtering followed by spectral filtering. Case C) Spectral filtering followed by spatial filtering. Table 6.2 shows the selected spatial filters and spectral filters and the classification accuracy of the subject A5 for left vs right task for all the three cases.

From Table 6.2, it can be seen that "Spectral followed by spatial filtering" achieves the best performance among all the three cases. In Case A, none of the spectral bands are eliminated and hence there is no effect of spectral filtering. Also the time taken is more, as the algorithm eliminates the spatial filters one by one with all the frequency bands intact. For Case B, it is seen that changing the order of spectral and spatial filtering has little effect on the performance. However, at the start, the total number of spatial filters is very high. Hence, performing spatial filtering before spectral filtering is computationally complex.

Next, the performances of S-IBCI and SS-IBCI are compared with other algorithms like Common Spatio-Spectral Pattern (CSSP) method [49], the Filter Bank CSP (FBCSP) method [92], the Discriminative Filter Bank CSP (DFBCSP) method [97], the Optimal Spatio-Spectral Filter Network (OSSFN) method [98] (OSSFNwFBCSP-OSSFN utilizing FBCSP, OSSFNwFBCSP-OSSFN utilizing DFBCSP) and the Bayesian Spatio-Spectral Filter Optimization (BSSFO) method [100]. Table 6.3 shows the classification performance (subject mean and standard deviation (nine subjects)) of all the algorithms for all the six classification tasks. It may be noted here that all the compared algorithms use the SVM as classifier. For all the algorithms except S-IBCI and SS-IBCI, the results have been adapted from [100]. From the Table 6.3, it can be observed that S-IBCI and SS-IBCI perform better than rest of the algorithms across all the tasks. In comparison to the time domain method CSSP, the proposed method achieves an average increment of 6-7%. In comparison to the filter bank based methods like the BCI competition winning FBCSP algorithm, it achieves an average increment of 11% with highest gain for foot vs tongue classification task. This might be due to efficient handling of non-stationarity in EEG signals. In comparison to the next best performing algorithm BSSFO, SS-IBCI achieves an average increment of 3%. However, BSSFO uses an ensemble of multiple classifiers from multiple bands in an iterative manner that makes the algorithm complex in nature, whereas the proposed approach uses a simpler approach employing fixed size bands with a single classifier. Also, the standard deviation of SS-IBCI is lower than BSSFO for all the tasks that shows that the proposed approach is more consistent across subjects in comparison to BSSFO. In the case of complex task like foot vs tongue the SS-IBCI achieves the highest increment of 5% over BSSFO. This shows the effectiveness of the proposed method in case of complex tasks.

Table 6.1: Classification performance (Subject Mean (SM) in % and Standard Deviation(SD)) of S-IBCI, SS-IBCI and bands obtained for each subject for all the six classification tasks for the BCI competition IV data set-IIa.

(a) Left vs right

Subject	Selected Bands	Testing Accuracy	
		S-IBCI	SS-IBCI
A1	1,2,5,6	96.53	96.53
A2	1,3,4,5,6	62.50	65.28
A3	1,6	98.61	99.31
A4	1,2,3,4,5,6	73.61	76.39
A5	2,4,6	81.25	83.33
A6	2,4	73.61	76.39
A7	3,4,5,6	81.25	83.33
A8	1,2,4	98.61	99.31
A9	2,4,5	93.06	93.75
SM		84.34	85.95
SD		13.03	12.01

(b) Left vs foot

Subject	Selected Bands	Testing Accuracy	
		S-IBCI	SS-IBCI
A1	1,2,4,5	100	100
A2	1,3,4,5,6	86.11	87.50
A3	1,2,3,6	95.83	95.83
A4	1,2,3,4,5	90.28	93.08
A5	1,2,3,4,5,6	78.47	83.33
A6	1,2,3,4,5,6	82.64	85.42
A7	3,4,5,6	100	100
A8	1,4,5,6	93.75	95.14
A9	1,2,4	97.92	97.92
SM		91.67	93.14
SD		7.81	6.28

(c) Left vs tongue

Subject	Selected Bands	Testing Accuracy	
		S-IBCI	SS-IBCI
A1	1,2,5	100	100
A2	2,3,5,6	78.47	78.47
A3	1,4,5,6	93.75	95.14
A4	1,3,5	93.75	93.75
A5	2,3,4,6	77.08	80.36
A6	1,2,3	79.86	81.94
A7	2,3,4,6	98.61	100
A8	1,2,3,4,5	97.22	97.92
A9	4,5,6	99.31	100
SM		90.89	91.98
SD		9.60	9.05

(d) Right vs foot

Subject	Selected Bands	Testing Accuracy	
		S-IBCI	SS-IBCI
A1	1,2,4	100	100
A2	3,6	81.94	82.64
A3	1,2,4,6	97.92	98.61
A4	1,3,4,5,6	94.44	94.44
A5	2,3,5,6	83.33	86.81
A6	3,4,5	81.25	81.25
A7	3,4,6	100	100
A8	1,2,3,4,5,6	96.53	96.53
A9	1,2,4	89.58	93.06
SM		91.67	92.59
SD		7.80	7.30

(e) Right vs tongue

Subject	Selected Bands	Testing Accuracy	
		S-IBCI	SS-IBCI
A1	1,4	100	100
A2	1,3,4,6	75	77.78
A3	1,2,4,5,6	99.31	100
A4	1,3,4	91.67	91.67
A5	1,3,4,5,6	90.97	90.97
A6	2,4,5,6	73.61	73.61
A7	2,3,4,5,6	99.31	99.31
A8	1,2,3,4,5	93.75	93.75
A9	1,2,3,4,5,6	90.97	90.97
SM		90.51	90.90
SD		9.90	9.46

(f) Foot vs tongue

Subject	Selected Bands	Testing Accuracy	
		S-IBCI	SS-IBCI
A1	1,2,4,6	80.56	83.33
A2	2,5,6	82.64	84.04
A3	2,3,4,5,6	90.97	94.44
A4	3,4,5	84.03	85.42
A5	2,3,5,6	65.97	70.83
A6	2,3,5	81.25	81.94
A7	2,3,4,5,6	87.50	88.19
A8	1,2,3,4,6	92.36	92.36
A9	1,3,6	95.83	96.53
SM		84.57	86.34
SD		8.76	7.78

Table 6.2: Classification performance in %, selected bands and spatial filters obtained for subject A5 for the left-right classification task for the BCI competition IV data set-IIa.

Order of Filter	Selected Bands	Selected Spatial filters	Testing Accuracy
CASE A	1,2,3, 4,5,6	$\mathbf{V}_{11}\mathbf{V}_{12}\mathbf{V}_{13}\mathbf{V}_{14}\mathbf{V}_{15}\mathbf{V}_{16}, \mathbf{V}_{21}\mathbf{V}_{22}\mathbf{V}_{23}\mathbf{V}_{24},$ $\mathbf{V}_{31}\mathbf{V}_{32}\mathbf{V}_{33}\mathbf{V}_{35}\mathbf{V}_{36}, \mathbf{V}_{41}\mathbf{V}_{42}\mathbf{V}_{43}\mathbf{V}_{44}$ $\mathbf{V}_{45}\mathbf{V}_{46}, \mathbf{V}_{41}\mathbf{V}_{42}\mathbf{V}_{43}\mathbf{V}_{44}\mathbf{V}_{45}\mathbf{V}_{46}, \mathbf{V}_{61}\mathbf{V}_{62}\mathbf{V}_{64}\mathbf{V}_{66}$	72.91
CASE B	4,5,6	$\mathbf{V}_{41}\mathbf{V}_{42}\mathbf{V}_{43}\mathbf{V}_{44}\mathbf{V}_{45}\mathbf{V}_{46},$ $\mathbf{V}_{41}\mathbf{V}_{42}\mathbf{V}_{43}\mathbf{V}_{44}\mathbf{V}_{45}\mathbf{V}_{46}, \mathbf{V}_{61}\mathbf{V}_{62}\mathbf{V}_{64}\mathbf{V}_{66}$	81.25
CASE C	2,4,6	$\mathbf{V}_{21}\mathbf{V}_{22}\mathbf{V}_{23}\mathbf{V}_{24}\mathbf{V}_{25},$ $\mathbf{V}_{43}\mathbf{V}_{44}\mathbf{V}_{45}\mathbf{V}_{46}, \mathbf{V}_{61}\mathbf{V}_{62}\mathbf{V}_{63}\mathbf{V}_{64}\mathbf{V}_{65}\mathbf{V}_{66}$	83.33

Table 6.3: Performance comparison of S-IBCI and SS-IBCI for the BCI competition IV data set-IIa.

Algorithms	left vs right	left vs foot	left vs tong.	right vs foot	right vs tong.	foot vs tong.
CCSP	79.78 $\pm$ 13.69	86.19 $\pm$ 12.47	84.80 $\pm$ 13.96	82.64 $\pm$ 14.09	86.34 $\pm$ 11.92	77.16 $\pm$ 9.55
FBCSP	76.31 $\pm$ 17.90	81.33 $\pm$ 13.18	80.86 $\pm$ 15.17	79.86 $\pm$ 15.52	80.32 $\pm$ 12.89	73.30 $\pm$ 11.17
DFBCSP	77.55 $\pm$ 18.33	80.71 $\pm$ 17.95	79.71 $\pm$ 18.28	82.10 $\pm$ 15.41	77.47 $\pm$ 15.27	73.30 $\pm$ 12.33
OSSFNwFBCSP	76.31 $\pm$ 18.59	80.94 $\pm$ 14.18	79.01 $\pm$ 11.02	75.54 $\pm$ 16.71	79.01 $\pm$ 10.31	75.08 $\pm$ 12.39
OSSFNwDFBCSP	75.62 $\pm$ 18.84	84.11 $\pm$ 12.15	78.32 $\pm$ 10.81	78.01 $\pm$ 15.38	79.55 $\pm$ 12.29	76.08 $\pm$ 10.11
BSSFO	83.80 $\pm$ 13.09	89.89 $\pm$ 10.71	90.43 $\pm$ 10.03	87.89 $\pm$ 11.29	89.04 $\pm$ 11.43	81.10 $\pm$ 9.73
S-IBCI	84.34 $\pm$ 13.03	91.67 $\pm$ 7.81	90.89 $\pm$ 9.60	91.67 $\pm$ 7.80	90.51 $\pm$ 9.90	84.57 $\pm$ 8.76
SS-IBCI	85.95 $\pm$ 12.01	93.14 $\pm$ 6.28	91.98 $\pm$ 9.05	92.59 $\pm$ 7.30	90.90 $\pm$ 9.46	86.34 $\pm$ 7.78

6.2.2 Performance Evaluation of S-IBCI and SS-IBCI using the BCI Competition III Data Set-IVa

The data set-IVa [18] consists of five subjects ('aa', 'al', 'av', 'aw', 'ay') and is recorded using one hundred eighteen electrodes. The subjects performed left hand, right hand and right foot motor-imagery tasks. In this study, the data for the right hand (class one) and foot (class two) motor-imagery tasks are used for classification.

Table 6.4 presents the number of train samples, test samples, selected bands and classification accuracies achieved for all the five subjects for both S-IBCI and SS-IBCI. It may be seen from this table that for the subject 'ay', SS-IBCI improves the performance of S-IBCI by 3% highlighting the importance of spatio-spectral filtering. Performances of both the proposed algorithms have been compared with CCSP, FBCSP, DFBCSP, OSSFN (OSSFNwFBCSP, OSSFNwDFBCSP) and BSSFO. Table 6.5 shows the performance for individual subjects and the mean accuracy and standard deviation for all the algorithms. The results of other algorithms except S-IBCI and SS-IBCI have been adapted from [100]

Table 6.4: Classification performance of S-IBCI, SS-IBCI and Bands obtained for each subject for the BCI competition III data set-IVa.

(a) Right hand vs foot

Subject	No. of Samples		Selected Bands	Testing Accuracy	
	Train	Test		S-IBCI	SS-IBCI
aa	168	112	2,4,6	86.61	87.50
al	224	56	2,3	100	100
av	84	196	1,4	73.57	75
aw	56	224	1,2,4,5,6	82.14	83.93
ay	28	252	1,4,6	86.11	89.29
SM				85.68	87.14
SD				9.55	9.05

where SVM has been used as the classifier. In case of subject ‘*aw*’, CSSP and BSSFO perform better than the proposed method. This might be due to use of time delay and ensemble of classifiers, respectively, however, on an average the proposed approach performs better than both of these algorithms. The proposed method performs significantly better than other similar methods like FBCSP (25-27%) and DFBCSP (24-26%) which use CSP as the baseline. This highlights the effectiveness of the proposed method.

Table 6.5: Performance comparison of S-IBCI and SS-IBCI on BCI competition III data set-IVa.

Algorithm	Subjects					SM $\pm$ SD
	aa	al	av	aw	ay	
CSSP	79.46	92.43	52.55	91.52	51.59	73.60 $\pm$ (20.33)
FBCSP	69.64	80.36	47.96	55.36	48.41	60.35 $\pm$ (14.21)
DFBCSP	69.64	82.14	54.08	50.89	48.41	61.03 $\pm$ (14.21)
OSSFNwFBCSP	75.00	83.93	53.06	74.11	48.81	66.98 $\pm$ (15.22)
OSSFNwDFBCSP	75.00	83.93	53.06	74.11	47.22	66.46 $\pm$ (15.93)
BSSFO	79.46	94.64	57.65	91.96	53.57	75.46 $\pm$ (19.06)
S-IBCI	86.61	100	73.57	82.14	86.11	85.68 $\pm$ (9.55)
SS-IBCI	87.50	100	75	83.93	89.29	87.14 $\pm$ (9.05)

The frequency bands of width 4 *Hz* employed in this chapter have been shown to yield a stable frequency response and provide the best results in comparison to other bands of varying widths in the range of 2-6 *Hz* [97, 194]. In contrast, freely searching of

the desired frequency bands will result in a higher computational overload. Therefore, the frequency bands of width 4 Hz have been selected in the range of 8-32 Hz . It is also possible to apply the proposed approach to datasets that comprise more than two classes either using a One-Versus-Rest or a Pair-Wise multi-classification approach.

6.3 Summary

A SRIT2NFIS based subject-specific spatio-spectral filter selection approach has been presented in this chapter for EEG based motor-imagery task classification in BCI. The proposed approach uses an interval type-2 fuzzy classifier, which can handle the non-stationarity in EEG signals and selects the appropriate spectral and spatio-spectral filters. The performances of S-IBCI and SS-IBCI have been evaluated using two publicly available BCI competition data sets, viz., data set-IVa [117] from the BCI competition III and data set-IIa [116] from the BCI competition IV. Based on the subject-wise studies, the experimental results indicate a significant performance improvement by both the spectral and spatio-spectral filter selection scheme. It also indicates presence of common frequency bands ((8-16) Hz and (20-32) Hz) for all subjects across all the classification tasks. The next chapter presents a realization of I-BCI for control of quadcopter movement.

Chapter 7

Realization of I-BCI for Control of a Quadcopter

In the Chapter 5, an Intelligent-BCI (I-BCI) system has been proposed with robust feature extraction algorithm and a self-regulated interval type-2 neuro-fuzzy inference system for handling the non-stationarity in EEG signals generated by motor-imagery tasks. In this chapter, a realization of the developed I-BCI system for control of quadcopter movement is presented. The quadcopter is stabilized using Proportional Integral Derivative (PID) control and altitude hold. The lateral motion of quadcopter is controlled by brain signal. It can adapt to the user and vice-versa. This feedback mechanism between the user and the quadcopter enables the user to control the device and adapt to its movement.

This chapter begins with a literature survey on BCI systems used for an external device control which is provided in Section 7.1. In Section 7.2, the architecture of the proposed system is presented in detail. The performance evaluation is presented in Section 7.3 and Section 7.4 provides the summary of this study.

7.1 A Review of BCI Systems for Device Control

In the literature, many BCI systems have been proposed for controlling external devices [195–197]. In [195], a non-invasive BCI system has been used to control the flight of a virtual helicopter. The subjects wore a sixty-four channel EEG cap that recorded sensorimotor rhythms from motor imagination. The left/right control signal from BCI2000 was

used to control the helicopter's movement. The subjects imagined moving their left hand to move the helicopter left and right hand to move it right. Initially, the helicopter moves forward with constant velocity and based on the imagination of the subjects, it rotates left or right. In [196], the BCI system has been further enhanced by adding a third control signal to modulate the helicopter's forward and backward movement. This system provides the users with continuous control of the virtual helicopter in three-dimensions. In [197], another BCI system has been employed to control the flight of a quadcopter, where human subjects were trained to modulate their sensorimotor rhythms to control a quadcopter navigating in a 3D physical space. The visual feedback was provided via a forward facing camera on the hull of the quadcopter. All these systems use a gel based EEG device with a large number of electrodes to record EEG data. Using such devices to record EEG data results in a longer preparation time and might cause discomfort to the user.

In order to overcome these issues, BCI systems have been proposed that use a low-cost portable EEG headset instead of using a clinic-used BCI headset [198, 199]. Recently, an EMOTIV Epoc device, a non-gel low-cost headset with fourteen electrodes and two reference channels has been used to control the flight of a quadcopter [198, 199]. In [198], a BCI system has been proposed that uses eye movements to control the flight of a quadcopter. In [199], facial expressions have been used and converted to user commands to fly a quadcopter. However, in this kind of device that has a limited number of channels, the non-stationarity present in the EEG data greatly influences the system [71, 91]. Also, if the subject suffers from serious motor impairment, it is not possible to use facial expressions and eye movements. Hence, there is a need to develop a mechanism that handles non-stationarity and enables such a device to control the movement of a quadcopter using brain signals.

7.2 Realization of I-BCI for Quadcopter Control

In this section, the architecture of the proposed system is presented and the individual components are explained. This system consists of a training or calibration phase and an online quadcopter control phase. In the training phase, data is collected from the subject

using an EMOTIV Epoc device. The raw data is then filtered using a fifth-order butterworth filter in the frequency range of 8-30 Hz . This frequency range is known to contain frequencies responsible for motor-imagery. The log-variance features are extracted from the data using the RoCSP algorithm as explained in Chapter 3. These features are then passed to the SRIT2NFIS algorithm that learns a classification model based on the data as explained in Chapter 4. The quadcopter control phase is shown in Fig. 7.1. Here, EEG data is collected from the subject in an online manner and classified using the learnt SRIT2NFIS classifier into left or right motor-imagery classes. Based on the classification of the signal, the quadcopter moves either left or right. The individual components of the proposed system are explained in detail in the next sections.

7.2.1 Data Collection and Pre-processing

The subject was in a sitting position and the EMOTIV Epoc device was fitted to the head of the subject. For the right motor-imagery task, the subject imagined catching a ball thrown towards him using his right hand. Similarly, for the left motor-imagery the subject imagined catching the ball using the left hand. A total of fifty trials for each left and right motor-imagery task is collected. Each of trial is about 2 seconds in duration. The training data set consists of thirty left and thirty right trials and the testing data set consists of twenty left and twenty right trials. The EEG signals were sampled at 128 Hz . The subject was monitored for movements like eye blink and facial muscle movement and reminded to minimize such actions. The collected trials are then filtered in a frequency range of 8-30 Hz to encompass the frequencies related to motor imagery tasks. Next, the details of EMOTIV Epoc device are presented.

7.2.1.1 EMOTIV Epoc Device

The proposed mechanism employs an EMOTIV Epoc headset, as shown in Fig. 7.1 to extract the EEG signals. The EMOTIV Epoc device (refer Fig. 7.2 adapted from [200]) consists of 16 bio-potential sensors. However, only 14-channel bio-potential sensors with gold-plated electrodes are used to capture the spatial resolution of the data. The remaining two are reference sensors. The 14 electrodes namely, AF3, AF4, F7, F8, F3, F4, FC5, FC6, T7, T8, P7, P8, O1 and O2 are based on the 10-20 international system

Figure 7.1: Mechanism for quadcopter flight control using the Intelligent-BCI system.

[201]. The frontal, central, parietal, occipital and temporal lobes are denoted by the letters F, C, P, O and T respectively. The electrodes are placed in pairs to capture the voltage difference between the neurons. The Epoc device uses a sampling rate of 128 Hz with 1.95-V least significant bit voltage resolution. Figure 7.3 shows the positions of the sensors on the user after the EMOTIV Epoc was fitted onto the user's head. It should be noted that the EMOTIV device is shifted to cover the motor cortex to capture the motor-imagery signals effectively.

Figure 7.2: EMOTIV headset.

7.2.2 Quadcopter Model

The quadcopter model used in this work is the AR.drone 2.0. (refer Fig. 7.4 adapted from [202]). It consists of a receiver module with wireless interface, a gyroscope unit, ultrasonic height sensor, accelerometer circuits, four propeller DC motors and a lithium-polymer battery. The quadcopter, often also referred to as quadrotor or drone, is a multicopter that has four equally-spaced rotors that propel and lift the quadcopter from the ground. There are two pairs of identical propellers (pusher/puller) that work in the

Figure 7.3: Positions of the shifted sensors on the subject

opposite directions. By regulating the speed of the individual rotors, the desired torque or turning power can be achieved. In this work, the altitude for the quadcopter has been fixed at 100 *cm* from the ground. Next, based on the left or right command provided to the quadcopter from the I-BCI system (as shown in Fig. 7.1), it moves to left or right by 50 *cm*. The attitude control has been implemented to move the quadcopter in lateral directions.

The control architecture shown in Fig. 7.5 governs the movement of the quadcopter. The reference input is an external left/right command (from the user) specified in terms of a desired roll angle. The error is computed as the difference between this reference input and the actual angle calculated based on attitude estimation from the on-board sensors. Subsequently, a PID controller computes the pulse width modulation mix for the four motors to achieve the desired roll. In other words, the controller minimizes the error towards zero. In the performed experiment, the left/ right commands are passed to the quadcopter as a desired roll angle of 15 degrees.

7.2.3 Experimental Setup

The above system is realized as follows: The subject was in a sitting position and the EMOTIV device was fitted onto the head of the subject. It was ensured that all the fourteen channel indicators are green indicating a good connection. The EEG data was

Figure 7.4: AR.drone 2.0.

Figure 7.5: Block diagram of quadcopter control system.

recorded and stored in the computer. Next, the features were extracted using the RoCSP algorithm and the SRIT2NFIS model was built in the calibration phase. In the online quadcopter control phase, the EEG signals are decoded using the I-BCI system and relayed to the quadcopter. A safety mechanism is realized such that, if the predicted class of a particular trial is not decoded with a high confidence (posterior probability of a prediction greater than 75%) , it is not relayed to the quadcopter. This mechanism is implemented by writing the classified commands in a buffer and the quadcopter application reads the commands from the buffer. Finally, the quadcopter moves based on the relayed commands. In this experiment, the computer ran a Windows 7 professional operating system with an Intel i5 2.60-*GHz* CPU.

7.3 Performance Evaluation

In this section, the performance of the proposed I-BCI system in controlling the quadcopter is evaluated. First, the I-BCI model is trained and evaluated using the collected data set. Next, in the online quadcopter control phase, the I-BCI system is used to control the movement of a quadcopter.

7.3.1 Performance Evaluation - Left-Right Motor-imagery Task Classification

The data set is recorded using fourteen electrodes. The subject performed two motor-imagery tasks (left and right) and a total of fifty left and fifty right trials are collected. Out of the hundred trials, sixty trials were used for building the model and remaining forty trials were used for testing. It should be noted that the I-BCI system does not eliminate any trial or channels in the calibration phase due to smaller channel availability. This also shows the good quality of the collected trials. The I-BCI system achieves a training accuracy of 100 % and a testing accuracy of 95 %. The results show that the I-BCI system is suitable for controlling the quadcopter due to its ability to provide good left-right classification accuracy.

7.3.2 Experimental Evaluation - Movement Control of Quadcopter

The performance of the proposed I-BCI system in controlling the quadcopter is evaluated. The objective of the experiment is to fly a quadcopter from point A to point B and fly back from point B to point A. The experiment requires the user to perform the left movements first, in order to move to point B starting from point A, followed by right movements to come back to point A from point B as shown in Fig. 7.6. In order to move from point A to point B, the quadcopter is required to make four consecutive left movements followed by four consecutive right movements to come back to its original position. This experiment is repeated ten times. Table 7.1 shows the patterns generated and the classification accuracies achieved by each of the runs. It can be seen from the Table 7.1 that the proposed system performs close to the expected output "Left Left Left Left - Right Right Right Right" on an average. In the case of "Run 6", it achieves the best performance of 100%. On an average, it achieves a classification performance of 83.75%.

Table 7.1: Performance evaluation (Mean and Standard Deviation (SD)), generated left/right patterns, accuracy for ten runs while performing the experiment.

Runs	Quadcopter Movement Patterns		Accuracy
Run 1	Left Left Left Left	Right Left Right Right	87.5
Run 2	Right Left Left Left	Right Right Right Right	87.5
Run 3	Left Left Right Left	Right Right Right Right	87.5
Run 4	Left Left Left Right	Right Left Right Right	75
Run 5	Left Right Left Left	Right Right Right Left	75
Run 6	Left Left Left Left	Right Right Right Right	100
Run 7	Left Right Left Left	Right Right Right Right	87.5
Run 8	Left Left Left Left	Left Right Right Left	75
Run 9	Left Left Right Left	Right Right Left Right	75
Run 10	Left Left Left Left	Right Left Right Right	87.5
Mean			83.75
SD			8.44

The starting position of the quadcopter is shown in Fig. 7.7. As the user thinks of left-hand motor-imagery task, the quadcopter starts moving left as shown in Fig. 7.8. It shows the images of the quadcopter at frames zero, thirty, sixty and ninety while moving left from point A to B.

Figure 7.6: Schematic diagram of the experiment. The user moves the quadcopter from point A to B and vice versa.

Figure 7.7: Quadcopter at the start position.

(a) Quadcopter at point A-(frame 0).

(b) Quadcopter moving towards point B-(frame 30).

(c) Quadcopter at point B-(frame 60).

(d) Quadcopter at point B-(frame 90).

Figure 7.8: The figure shows the left movement of quadcopter at four time instances while moving from point A to point B.

Once the quadcopter has reached the point B, the subject started thinking of right hand motor-imagery task and the quadcopter starts moving right. Similar to the Fig. 7.8, the Fig. 7.9 shows the movement of quadcopter while moving from position B, back to position A. As seen from Figs. 7.8 and 7.9, the user is able to control the movement of quadcopter for the desired task.

The proposed system is only realized for left-right movement control of quadcopter. It is not realized for up/down movement as the employed EEG feature extraction algorithm can only generate features for two classes of EEG data and realizing all the movements (left/right/top/down/turn) simultaneously requires multi-class features. The current system also needs to be calibrated before use, which can sometimes cause discomfort to the user.

(a) Quadcopter at point B-(frame 100).

(b) Quadcopter moving towards point A-(frame 130).

(c) Quadcopter moving towards point A-(frame 160).

(d) Quadcopter at point A-(frame 200).

Figure 7.9: The figure shows the right movement of quadcopter at four time instances while moving back from point B to point A.

7.4 Summary

This chapter presents a realization of I-BCI system to control the left-right movement of a quadcopter using a non-invasive BCI device. The BCI device used to record raw EEG data in this work is the fourteen channel EMOTIV Epoc and the quadcopter used is the AR.drone 2.0. The RoCSP algorithm is used for feature extraction and the SRIT2NFIS classifier is employed to classify the signals into left/right classes. An experiment of flying the quadcopter to and fro from a point shows that the user is successfully able to control the quadcopter.

Chapter 8

Conclusions and Future Directions

8.1 Thesis Summary

This thesis develops a unified BCI system that provides improved classification accuracy by addressing the key problems of robust feature extraction (handle artifacts and eliminate redundant channels), non-stationarity in EEG data and spatio-spectral filter selection. The major contributions of this work are summarized as follows:

- A Robust Common Spatial Pattern (RoCSP) feature extraction algorithm that provides robust features for classification. It eliminates trials affected by artifacts and discards redundant channels to improve the classification performance of the BCI system.
- A Self-Regulated Interval Type-2 Neuro-Fuzzy Inference System (SRIT2NFIS) that is capable of handling non-stationarity in the data. This system handles non-stationarity in data by modeling it as uncertainty using type-2 fuzzy sets.
  - A self-regulating sequential learning algorithm that decides whether a particular sample should be used for rule growing/updating the parameters or discarded without learning or reserved to be considered at a later stage.
  - A computationally efficient type-reduction algorithm that converts the interval type-1 fuzzy set to a fuzzy number for a particular rule.
  - A regularized projection based learning algorithm that computes optimal weights of the network by minimizing the sum of squared hinge-error.

- An Intelligent-Brain Computer Interface (I-BCI) system that combines RoCSP and SRIT2NFIS to address the key problem of non-stationarity in EEG based BCI systems.
- A mechanism for selecting subject-specific spatio-spectral filters.
- Realization of I-BCI for control of a quadcopter.

Each of the contributions is summarized below.

8.1.1 Robust Common Spatial Pattern Feature Extraction Algorithm

A Robust Common Spatial Pattern (RoCSP) algorithm has been proposed to provide robust features for classification by eliminating the trials affected by artifacts and discarding redundant channels. Trial selection has been carried out using z -scores to remove the trials with high intratask variation. Further, channel selection has been carried out using Cohen's- d effect-size to discard the channels with minimal intertask variation. These selected trials and channels are then used to determine the spatial filters that lead to robust log-variance features for classification.

The performance of RoCSP was evaluated on two benchmark BCI competition data sets (Data sets IVa and IIa from BCI competition III and IV respectively) and compared with existing state-of-the-art feature extraction algorithms. Extensive experiments involving six classification tasks (left vs right hand classification, left vs foot classification, left vs tongue classification, right vs foot classification, right vs tongue classification and foot vs tongue classification) have been conducted using BCI competition data set-IIa. The subject-wise comparison with the baseline CSP algorithm on these classification tasks showed better performance of RoCSP. Further, in comparison to existing algorithms (GLRCSP, CCSP1, CCSP2, DLCSPauto, DLCSPcv, DLCSPcvdiff, SSRCSPP, TRCSP, WTRCSP and SRCSP), subject-wise results showed that RoCSP performed better even though most of these algorithms use additional information like location of electrodes, data from other subjects etc. to regularize the covariance matrices. In case of BCI competition III-data set IVa, RoCSP outperformed the CSP algorithm by 16% using only 17

channels on an average compared to all the 118 channels used by the CSP algorithm. In comparison to the other above mentioned state-of-the-art algorithms, RoCSP performed better even though it employed lesser number of channels. An important observation is that RoCSP might require all the channels in cases where the channel availability is low. However, the reduction in number of channels is significant with increasing channels.

8.1.2 Self-Regulated Interval Type-2 Neuro-Fuzzy Inference System

A self-regulated interval type-2 neuro-fuzzy inference system has been proposed to handle the non-stationary data. It handles the non-stationarity in the input features by modeling it as uncertainty in the classifier. The structure of SRIT2NFIS is a five-layered modified Takagi-Sugeno-Kang interval type-2 fuzzy inference mechanism and the learning algorithm is a self-regulatory learning mechanism that estimates the rules and associated parameters from the incoming data.

The performance of SRIT2NFIS has been evaluated on a set of benchmark classification problems from the UCI machine learning repository. The performance comparison with existing type-1 algorithms including eClass, OS-F-ELM, SAFIS, McFIS, and SVM showed that the use of interval type-2 sets in SRIT2NFIS enabled it to perform better using lesser number of rules. Further, the comparison with other type-2 systems, including T2SONFS, MRITNFS-WB(A) and IT2NFC-SMM on three binary classification problems (SVMguide 1, Heart-s and Tao) showed improved classification accuracy of SRIT2NFIS, thereby proving its effectiveness.

8.1.3 Intelligent-Brain Computer Interface for Handling Non-stationarities in EEG Signals

An Intelligent-BCI (I-BCI) system combines the RoCSP feature extraction algorithm and the self-regulated interval type-2 neuro fuzzy inference system to classify the EEG signals and handle non-stationarity. The SRIT2NFIS classifier handles the non-stationary features generated by the RoCSP algorithm as uncertainty using interval type-2 fuzzy sets in the antecedent of fuzzy rules. The performance of I-BCI in handling the non-stationarity in EEG signals generated by motor-imagery tasks in inter/intrasession settings has been

evaluated using two publicly available BCI competition data sets. The performance comparison showed that I-BCI outperformed the existing algorithms by 3-5% in simple tasks and 6-8% in complex tasks. The results also showed that I-BCI improvements were more significant for subjects with lower CSP performance. A Friedman test followed by post-hoc Bonferroni-Dunn rank test proved that the performance improvement of I-BCI over the other algorithms (eleven algorithms) is statistically significant.

8.1.4 Discriminative Subject-Specific Spatio-Spectral Filter Selection Approach for EEG based Motor-Imagery Task Classification

A SRIT2NFIS based subject-specific spatio-spectral filter selection approach has been presented for EEG based motor-imagery task classification in BCI. This approach uses an interval type-2 fuzzy classifier, which can handle the non-stationarity in EEG signals. Further, the proposed approach selects the appropriate spectral and spatio-spectral filters. Performances of S-IBCI and SS-IBCI have been evaluated using two publicly available BCI competition data sets, viz., data set-IVa [117] from the BCI competition III and data set-IIa [116] from the BCI competition IV. Based on the subject-wise studies, the experimental results indicated a significant improvement in performance by both the spectral and spatio-spectral filter selection scheme. It also indicated the presence of common frequency bands ((8-16) *Hz* and (20-32) *Hz*) for all subjects across all the classification tasks. It can be concluded that spectral filtering is more important than spatial filtering and order of filtering has little impact on the performance. Further, it outperformed other algorithms like FBCSP, DFBCSP and BSSFO.

8.1.5 Realization of I-BCI for Control of Quadcopter

The I-BCI system has been realized for left-right movement control of quadcopter. An EMOTIV Epoc is used for data recording and AR.drone 2.0 is used as the control device. An I-BCI based framework converts the raw EEG signals into user commands for quadcopter movement control. The decoded intentions of the user (move left, move right) are relayed to the quadcopter over a wireless connection. The performance evaluation in flying the quadcopter (fly to and fro from a point) confirmed that the user is able to control the quadcopter successfully with good accuracy.

8.2 Future Directions

Based on the studies done in this thesis, some possible research directions to further extend this work are given below.

8.2.1 Learning Algorithm

- **Adapting Center and Width of SRIT2NFIS**

In SRIT2NFIS algorithm, the rule centers and widths are fixed after being learnt from the data. However, for time varying data, it would be ideal if the rule centers and widths could also be adapted based on the data. A preliminary work has been done in [154] where the centers and widths are adapted based on the incoming data. The structure of the system employs interval type-2 fuzzy sets in the antecedent and the fuzzy rules are of the TSK type. The parameters of the network, including center and widths are adapted by employing an Extended Kalman Filter (EKF) based learning algorithm. This algorithm, however, is not suitable for classification problems as it lacks a class specific criterion. As a part of future work, this algorithm should be adapted for BCI applications.

- **Cost Functions**

SRIT2NFIS algorithm employs a hinge loss error based cost function. In the literature, various other cost functions like entropy based cost function, logarithmic error based cost function have been proposed. The entropy based cost function has been shown to constrain all the moments of error, unlike mean squared error, which only minimizes based on second moment of error. In [203], a neuro-fuzzy inference system the error entropy minimization has been proposed. These cost functions have to be studied and can be integrated with SRIT2NFIS for better performance.

8.2.2 Fuzzy Inference Mechanism

- **Membership Functions**

In the literature, other membership functions such as the elliptical membership function, Cauchy membership functions etc. have been presented. These membership functions have different advantages depending on the problem setting. In

the literature, a q Gaussian function has been proposed recently which can realize various radial basis functions based on different values of q [204]. This membership function should be studied to improve the existing algorithm.

- **Inference Mechanism**

The proposed system employs the Takagi-Sugeno-Kang fuzzy inference mechanism. In the literature, other inference mechanisms have been proposed which emphasize on rule interpretability [164]. There is a need to develop an inference mechanism that achieves the dual purpose of interpretability and accuracy. The self-regulating learning algorithm should be able to choose the inference mechanism based on the data and requirement.

- **Modeling Non-stationarity**

In this thesis, the non-stationarity is modeled using uncertain mean with fixed standard deviation. In the literature, other mechanisms such as the uncertain standard deviation with fixed mean have been proposed [205]. As a part of future work, the effect of this mechanism can be analyzed. The self-regulating learning algorithm should be able to select the right mechanism based on incoming data.

- **Handling High Dimensional Data**

One of the existing issues with the current fuzzy inference system is handling high dimension of input features. As the dimension of input features increases, the net membership value of a rule decreases leading to lower performance. Hence, there a need to improve the current inference system which has the capability of handling high dimensional data.

8.2.3 BCI Systems

- **Calibration Free BCI**

In the proposed BCI system, it is essential to train the user and calibrate the system before using it. This requires a longer waiting time before the BCI system is ready for use to a satisfactory level and causes discomfort to the user. The calibration

phase usually requires 20-30 minutes before the system becomes operational. Hence, it is necessary to improve the BCI system such that it operates efficiently in an unsupervised and online manner.

- **Multi-class BCI**

The current BCI system is used to analyze only two classes of motor imagery tasks. It has been shown in this thesis that the developed SRIT2NFIS classifier has the ability to perform multi-class classification. The current BCI system can be extended to handle multi-class BCI data. This would require the feature extraction to be extended to produce features for multiple classes simultaneously. Such a system would be able to provide additional degree of freedom to the subject, enabling him to use multiple commands to control the external devices.

- **Subject-specific Time-segment Extraction**

The EEG signal processing is performed in the temporal, spatial and spectral domains. The proposed BCI provides spectral and spatial filters for each of the individual subjects. It assumes a common and uniform time segment for every subject. However, the time segment can vary for different subjects. The proposed BCI system can be extended to incorporate subject-specific time segment extraction.

- **Adapting the Frequency Bands**

The proposed BCI system selects bands from a predefined filter bank to find subject-specific spatio-spectral filters. However, the widths of the bands are fixed. Adapting the width of the bands might lead to a better performance. The current BCI system could be improved by devising a mechanism to freely search for the optimal frequency bands and their corresponding width.

Appendix

There are a total of nine parameters in our algorithm namely: Delete threshold (E_d), Add threshold (E_a), Update threshold (E_l), Novelty threshold (E_s), Gamma (γ), Intra-class overlap factor (κ), Inter-class overlap factor (η), Pruning threshold (E_p) and Pruning window (E_w). Even though our algorithm has nine parameters, only four of them (Add threshold, Update threshold, Novelty threshold, Inter-class overlap factor) are sensitive and need to be tuned for each data set. The rest of the parameters (Delete threshold, Gamma, Intra-class overlap factor, Pruning threshold and Pruning window) can be kept constant for all the data sets. The tunable parameters are selected using five-fold cross-validation with training data alone. The influence of these parameter values on the test performance has been shown using the IRIS dataset in Fig. 8.1. It can be seen from the graphs that the classification accuracy does not vary much with change of parameter values. It can be concluded that the performance of proposed algorithm is not sensitive to these parameters.

Further, we describe the rule growing and pruning using an example to show the structure learning process of the SRIT2NFIS. We use the IRIS dataset to illustrate the rule history in Fig. 8.2. The rule history is plotted against the samples used in training samples. It shows the samples for which the rules are plotted and pruned. It can be seen from the figure that initially more number of rules are added and at a later stage these rules are fine tuned. Rule pruning takes place when contribution of a rule is less than a pruning threshold for consequent samples. The figure shows the rule pruning for the IRIS dataset.

(a) Influence of Add threshold

(b) Influence of eta

(c) Influence of Novelty threshold

(d) Influence of Update threshold

Figure 8.1: Influence of tunable parameters on the IRIS classification accuracy.

Figure 8.2: Rule growth history for IRIS dataset.

Publications List

Journals

1. A.K. Das, K. Subramanian, S. Suresh: “An evolving interval type-2 neuro-fuzzy inference system and its meta-cognitive sequential learning algorithm”, *IEEE Transactions on Fuzzy Systems*, Vol. 23, No. 6, pp. 2080-2093, 2015.
2. A.K. Das, S. Suresh, N. Sundararajan: “A self-regulated interval type-2 neuro-fuzzy inference system for handling non-stationarities in EEG signals for BCI”, *IEEE Transactions on Fuzzy Systems*, DOI: 10.1109/TFUZZ.2016.2540072.
3. A.K. Das, Nguyen Anh, S. Suresh, N. Srikanth: “An interval type-2 fuzzy inference system and its meta-cognitive learning algorithm”, *Evolving Systems*, Vol. 7, No. 2, pp. 95-105, 2016.
4. K. Subramanian, A.K. Das, S. Suresh, R. Savitha: “A meta-cognitive interval type-2 fuzzy inference system and its projection based learning algorithm”, *Evolving Systems*, Vol. 5, No. 4, pp. 219-230, 2014.
5. A.K. Das, S. Suresh, N. Sundararajan: “A discriminative subject-specific spatio-spectral filter selection approach for EEG based motor-imagery task classification”, *Expert Systems With Applications*, Vol. 64, pp. 375-384, 2016.

Conference Proceedings

1. A.K. Das, S. Suresh, N. Srikanth: “Meta-cognitive interval type-2 neuro-fuzzy inference system for wind prediction”, *IEEE International Conference on Multi-sensor Fusion and Information Integration for Intelligent Systems (MFI)*, pp. 1-6, 2014.

2. A.K. Das, K. Subramanian, S. Suresh: “A computationally fast interval type-2 neuro-fuzzy inference system and its meta-cognitive projection based learning algorithm”, *IEEE International Joint Conference on Neural Networks (IJCNN)*, pp. 1510-1516, 2014.
3. A.K. Das, S. Suresh: “An effect-size based channel selection algorithm for mental task classification in brain computer interface”, *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 3140-3145, 2015.
4. A.K. Das, S. Suresh, N. Sundararajan, K. Subramanian: “A subject-specific frequency band selection for efficient BCI- An interval type-2 fuzzy inference system approach”, *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1-8, 2015.
5. A.K. Das, S. Suresh, N. Sundararajan: “A fully tuned sequential interval type-2 fuzzy inference system for motor-imagery task classification”, *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2016.
6. A.K. Das, S. Suresh, N. Sundararajan, Teck Leong Tan: “Meta-cognitive interval type-2 fuzzy controller for quadcopter flight control- An EEG based approach”, *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2016.
7. A.K. Das, S. Suresh, N. Sundararajan: “A robust interval type-2 fuzzy inference based BCI system”, *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2016.

Bibliography

- [1] T. O. Nelson and L. Narens. Metamemory: a theoretical framework and new findings. *The Psychology of Learning and Motivation*, 26:125–141, 1990.
- [2] A. Souza, A. Kelleher, R. Cooper, R. A. Cooper, L. I. Iezzoni, and D. M. Collins. Multiple sclerosis and mobility-related assistive technology: systematic review of literature. *Journal of rehabilitation research and development*, 47(3):213, 2010.
- [3] M. Betke, J. Gips, and P. Fleming. The camera mouse: visual tracking of body features to provide computer access for people with severe disabilities. *IEEE Trans. Neural Systems and Rehabilitation Engineering*, 10(1):1–10, 2002.
- [4] H. Zhou and H. Hu. Human motion tracking for rehabilitation: A survey. *Biomedical Signal Processing and Control*, 3(1):1–18, 2008.
- [5] E. A. Curran and M. J. Stokes. Learning to control brain activity: a review of the production and control of EEG components for driving brain–computer interface (BCI) systems. *Brain and Cognition*, 51(3):326–336, 2003.
- [6] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan. Brain–computer interfaces for communication and control. *Clinical Neurophysiology*, 113(6):767–791, 2002.
- [7] G. Pfurtscheller and C. Neuper. Motor imagery and direct brain–computer communication. *Proceedings of the IEEE*, 89(7):1123–1134, 2001.
- [8] J. R. Wolpaw, G. E. Loeb, B. Z. Allison, E. Donchin, O. F. D. Nascimento, W. J. Heetderks, F. Nijboer, W. G. Shain, and J. N. Turner. BCI meeting 2005-workshop on signals and recording methods. *IEEE Trans. Neural Systems and Rehabilitation Engineering*, 14(2):138–141, 2006.

- [9] M. A. Lebedev and M. A. Nicolelis. Brain-machine interfaces: past, present and future. *TRENDS in Neurosciences*, 29(9):536–546, 2006.
- [10] L. R. Hochberg, M. D. Serruya, G. M. Friehs, J. A. Mukand, M. Saleh, A. H. Caplan, A. Branner, D. Chen, R. D. Penn, and J. P. Donoghue. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, 442(7099):164–171, 2006.
- [11] A. B. Schwartz, X. T. Cui, D. J. Weber, and D. W. Moran. Brain-controlled interfaces: movement restoration with neural prosthetics. *Neuron*, 52(1):205–220, 2006.
- [12] J.-D. Haynes and G. Rees. Decoding mental states from brain activity in humans. *Nature Reviews Neuroscience*, 7(7):523–534, 2006.
- [13] F. Malouin, C. L. Richards, P. L. Jackson, F. Dumas, and J. Doyon. Brain activations during motor imagery of locomotor-related tasks: A pet study. *Human brain mapping*, 19(1):47–62, 2003.
- [14] J. Mellinger, G. Schalk, C. Braun, H. Preissl, W. Rosenstiel, N. Birbaumer, and A. Kübler. An meg-based brain-computer interface (bci). *Neuroimage*, 36(3):581–593, 2007.
- [15] K. Jahn, A. Deutschländer, T. Stephan, M. Strupp, M. Wiesmann, and T. Brandt. Brain activation patterns during imagined stance and locomotion in functional magnetic resonance imaging. *Neuroimage*, 22(4):1722–1731, 2004.
- [16] K. INOUE, D. Mori, K. Sugioka, G. Pfurtscheller, and K. Kumamaru. Feature extraction of EEG signals during right and left motor imagery. In *SICE annual conference*, pages 2183–2187, 2004.
- [17] H. Ramoser, J. Muller-Gerking, and G. Pfurtscheller. Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE Trans. Rehabilitation Engineering*, 8(4):441–446, 2000.

- [18] B. Blankertz, K.-R. Müller, D. J. Krusienski, G. Schalk, J. R. Wolpaw, A. Schlögl, G. Pfurtscheller, J. R. Millan, M. Schröder, and N. Birbaumer. The BCI competition iii: validating alternative approaches to actual BCI problems. *IEEE Trans. Neural Systems and Rehabilitation Engineering*, 14(2):153–159, 2006.
- [19] M. Kawanabe and C. Vidaurre. Improving BCI performance by modified common spatial patterns with robustly averaged covariance matrices. In *World Congress on Medical Physics and Biomedical Engineering*, pages 279–282. Springer, 2009.
- [20] C. Sannelli, M. Braun, and K.-R. Müller. Improving BCI performance by task-related trial pruning. *Neural Networks*, 22(9):1295–1304, 2009.
- [21] X. Yong, , R. K. Ward, and G. E. Birch. Robust common spatial patterns for EEG signal preprocessing. In *Intl. Conf. of the IEEE Engineering in Medicine and Biology Society*, pages 2087–2090. IEEE, 2008.
- [22] N. Tomida, T. Tanaka, S. Ono, M. Yamagishi, and H. Higashi. Active data selection for motor imagery EEG classification. *IEEE Trans. Biomedical Engineering*, 62(2):458–467, 2015. doi: 10.1109/TBME.2014.2358536.
- [23] J. R. Wolpaw, N. Birbaumer, W. J. Heetderks, D. J. McFarland, P. H. Peckham, G. Schalk, E. Donchin, L. A. Quatrano, C. J. Robinson, and T. M. Vaughan. Brain-computer interface technology: a review of the first international meeting. *IEEE Trans. Rehabilitation Engineering*, 8(2):164–173, 2000.
- [24] M. Krauledat, G. Dornhege, B. Blankertz, and K.-R. Müller. Robustifying EEG data analysis by removing outliers. *Chaos and Complexity Letters*, 2(3):259–274, 2007.
- [25] P. Shenoy, M. Krauledat, B. Blankertz, R. P. Rao, and K.-R. Müller. Towards adaptive classification for BCI. *Journal of Neural Engineering*, 3(1):R13, 2006.
- [26] C. Tangwiriyaakul, R. Verhagen, M. J. van Putten, and W. L. Rutten. Importance of baseline in event-related desynchronization during a combination task of motor imagery and motor observation. *Journal of Neural Engineering*, 10(2):026009, 2013.

- [27] J. M. Mendel and X. Liu. New closed-form solutions for Karnik-Mendel algorithm+ defuzzification of an interval type-2 fuzzy set. In *IEEE Intl. Conf. on Fuzzy System*, pages 1–8. IEEE, 2012.
- [28] M. Nie and W. W. Tan. Towards an efficient type-reduction method for interval type-2 fuzzy logic systems. In *Intl. Conf. on Fuzzy Systems.(IEEE World Congress on Computational Intelligence)*., pages 1425–1432. IEEE, 2008.
- [29] G. S. Babu and S. Suresh. Sequential projection-based metacognitive learning in a radial basis function network for classification problems. *IEEE Trans. Neural Networks and Learning Systems*, 24(2):194–206, 2013.
- [30] D. Zhang, Y. Wang, X. Gao, B. Hong, and S. Gao. An algorithm for idle-state detection in motor-imagery-based brain-computer interface. *Computational Intelligence and Neuroscience*, 2007:5–5, 2007.
- [31] G. Pfurtscheller, C. Neuper, D. Flotzinger, and M. Pregenzer. EEG-based discrimination between imagination of right and left hand movement. *Electroencephalography and clinical Neurophysiology*, 103(6):642–651, 1997.
- [32] D. J. McFarland and J. R. Wolpaw. EEG-based communication and control: speed–accuracy relationships. *Applied Psychophysiology and Biofeedback*, 28(3):217–231, 2003.
- [33] D. J. McFarland, W. A. Sarnacki, T. M. Vaughan, and J. R. Wolpaw. Brain-computer interface (BCI) operation: signal and noise during early training sessions. *Clinical Neurophysiology*, 116(1):56–62, 2005.
- [34] J. R. Wolpaw and D. J. McFarland. Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans. *Proceedings of the National Academy of Sciences of the United States of America*, 101(51):17849–17854, 2004.
- [35] G. Townsend, B. Graimann, and G. Pfurtscheller. A comparison of common spatial patterns with complex band power features in a four-class BCI experiment. *IEEE Trans. Biomedical Engineering*, 53(4):642–651, 2006.

- [36] J. d. R. Millán and J. Mouriño. Asynchronous BCI and local neural classifiers: an overview of the adaptive brain interface project. , *IEEE Trans. Neural Systems and Rehabilitation Engineering*, 11(2):159–161, 2003.
- [37] C. Vidaurre, N. Krämer, B. Blankertz, and A. Schlögl. Time domain parameters as a feature for EEG-based brain–computer interfaces. *Neural Networks*, 22(9):1313–1319, 2009.
- [38] R. Boostani and M. H. Moradi. A new approach in the BCI research based on fractal dimension as feature and Adaboost as classifier. *Journal of Neural Engineering*, 1(4):212, 2004.
- [39] A. Schlögl, K. Lugger, and G. Pfurtscheller. Using adaptive autoregressive parameters for a brain-computer-interface experiment. In *International Conference of the IEEE on Engineering in Medicine and Biology Society*, pages 1533–1535. IEEE, 1997.
- [40] M. Kaper, P. Meinicke, U. Grosse-kathoefer, T. Lingner, and H. Ritter. BCI competition 2003-data set Iib: support vector machines for the P300 speller paradigm. *IEEE Trans. Biomedical Engineering*, 51(6):1073–1076, 2004.
- [41] U. Hoffmann, G. Garcia, J.-M. Vesin, K. Diserens, and T. Ebrahimi. A boosting approach to P300 detection with application to brain-computer interfaces. In *International IEEE EMBS Conference on Neural Engineering*, pages 97–100. IEEE, 2005.
- [42] T. Wang, J. Deng, and B. He. Classifying EEG-based motor imagery tasks by means of time–frequency synthesized spatial patterns. *Clinical Neurophysiology*, 115(12):2744–2753, 2004.
- [43] T. Wang and B. He. An efficient rhythmic component expression and weighting synthesis strategy for classifying motor imagery EEG in a brain–computer interface. *Journal of neural engineering*, 1(1):1, 2004.

- [44] N. Yamawaki, C. Wilke, Z. Liu, and B. He. An enhanced time-frequency-spatial approach for motor imagery classification. , *IEEE Trans. Neural Systems and Rehabilitation Engineering*, 14(2):250–254, 2006.
- [45] D. Coyle, G. Prasad, and T. M. McGinnity. A time-frequency approach to feature extraction for a brain-computer interface with a comparative analysis of performance measures. *EURASIP journal on applied signal processing*, 2005:3141–3151, 2005.
- [46] B.-h. Yang, G.-z. Yan, R.-g. Yan, and T. Wu. Adaptive subject-based feature extraction in brain-computer interfaces using wavelet packet best basis decomposition. *Medical engineering & physics*, 29(1):48–53, 2007.
- [47] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K.-R. Muller. Optimizing spatial filters for robust EEG single-trial analysis. *Signal Processing Magazine, IEEE*, 25(1):41–56, 2008.
- [48] D. J. McFarland, L. M. McCane, S. V. David, and J. R. Wolpaw. Spatial filter selection for EEG-based communication. *Electroencephalography and clinical Neurophysiology*, 103(3):386–394, 1997.
- [49] S. Lemm, B. Blankertz, G. Curio, and K.-R. Muller. Spatio-spectral filters for improving the classification of single trial EEG. *IEEE Trans. Biomedical Engineering*, 52(9):1541–1548, 2005.
- [50] K. Fukunaga. *Introduction to statistical pattern recognition*. Academic press, 1990.
- [51] Z. J. Koles. The quantitative extraction and topographic mapping of the abnormal components in the clinical eeg. *Electroencephalography and clinical Neurophysiology*, 79(6):440–447, 1991.
- [52] Z. J. Koles, J. Lind, and A. Soong. Spatio-temporal decomposition of the EEG: a general approach to the isolation and localization of sources. *Electroencephalography and clinical Neurophysiology*, 95(4):219–230, 1995.

- [53] Z. J. Koles and A. C. Soong. EEG source localization: implementing the spatio-temporal decomposition approach. *Electroencephalography and clinical Neurophysiology*, 107(5):343–352, 1998.
- [54] J. Müller-Gerking, G. Pfurtscheller, and H. Flyvbjerg. Designing optimal spatial filters for single-trial EEG classification in a movement task. *Clinical neurophysiology*, 110(5):787–798, 1999.
- [55] K.-R. Müller, M. Krauledat, G. Dornhege, G. Curio, and B. Blankertz. Machine learning techniques for brain-computer interfaces. *Biomedical Engineering*, pages 11–22, 2004.
- [56] C. Tangwiriyaikul, V. Mocioiu, M. J. van Putten, and W. L. Rutten. Classification of motor imagery performance in acute stroke. *Journal of Neural Engineering*, 11(3):036001, 2014.
- [57] S. R. Liyanage, C. Guan, H. Zhang, K. K. Ang, J. Xu, and T. H. Lee. Dynamically weighted ensemble classification for non-stationary EEG processing. *Journal of Neural Engineering*, 10(3):036007, 2013.
- [58] C. Sannelli, C. Vidaurre, K.-R. Müller, and B. Blankertz. CSP patches: an ensemble of optimized spatial filters. an evaluation study. *Journal of Neural Engineering*, 8(2):025012, 2011.
- [59] M. Grosse-Wentrup, C. Liefhold, K. Gramann, and M. Buss. Beamforming in non-invasive brain-computer interfaces. *IEEE Transactions on Biomedical Engineering*, 56(4):1209–1219, 2009.
- [60] M. Grosse-Wentrup, K. Gramann, and M. Buss. Adaptive spatial filters with pre-defined region of interest for EEG based brain-computer-interfaces. In *Advances in neural information processing systems*, pages 537–544, 2006.
- [61] H. Wang and W. Zheng. Local temporal common spatial patterns for robust single-trial EEG classification. *IEEE Trans. Neural Systems and Rehabilitation Engineering*, 16(2):131–139, 2008.

- [62] H. Wang and D. Xu. Comprehensive common spatial patterns with temporal structure information of EEG data: minimizing nontask related EEG component. *IEEE Trans. Biomedical Engineering*, 59(9):2496–2505, 2012.
- [63] H. Wang. Discriminant and adaptive extensions to local temporal common spatial patterns. *Pattern Recognition Letters*, 34(10):1125–1129, 2013.
- [64] C. Gouy-Pailler, M. Congedo, C. Brunner, C. Jutten, and G. Pfurtscheller. Nonstationary brain source separation for multiclass motor imagery. *IEEE Trans. Biomedical Engineering*, 57(2):469–478, 2010.
- [65] T.-H. Nguyen, S.-M. Park, K.-E. Ko, and K.-B. Sim. Multi-class stationary CSP for optimal feature separation of brain source in bci system. In *Intl. Conf. on Control, Automation and Systems (ICCAS)*, pages 1035–1039. IEEE, 2012.
- [66] M. Grosse-Wentrup and M. Buss. Multiclass common spatial patterns and information theoretic feature extraction. *IEEE transactions on Biomedical Engineering*, 55(8):1991–2000, 2008.
- [67] H. Wang. Harmonic mean of Kullback–Leibler divergences for optimizing multiclass EEG spatio-temporal filters. *Neural processing letters*, 36(2):161–171, 2012.
- [68] 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 Trans. Biomedical Engineering*, 60(8):2123–2132, 2013.
- [69] E. A. Mousavi, J. J. Maller, P. B. Fitzgerald, and B. J. Lithgow. Wavelet common spatial pattern in asynchronous offline brain computer interfaces. *Biomedical Signal Processing and Control*, 6(2):121–128, 2011.
- [70] O. Falzon, K. P. Camilleri, and J. Muscat. The analytic common spatial patterns method for EEG-based BCI data. *Journal of neural engineering*, 9(4):045009, 2012.
- [71] F. Lotte and C. Guan. Regularizing common spatial patterns to improve BCI designs: unified theory and new algorithms. *IEEE Trans. Biomedical Engineering*, 58(2):355–362, 2011.

- [72] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.
- [73] Y. Li and C. Guan. An extended EM algorithm for joint feature extraction and classification in brain-computer interfaces. *Neural Computation*, 18(11):2730–2761, 2006.
- [74] S. Sun and C. Zhang. Adaptive feature extraction for EEG signal classification. *Medical and Biological Engineering and Computing*, 44(10):931–935, 2006.
- [75] H. Kang, Y. Nam, and S. Choi. Composite common spatial pattern for subject-to-subject transfer. *IEEE Signal Processing Letters*, 16(8):683–686, 2009.
- [76] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. Regularized common spatial patterns with generic learning for EEG signal classification. In *Intl. Conf. of the IEEE in Engineering in Medicine and Biology Society*, pages 6599–6602. IEEE, 2009.
- [77] B. Blankertz, M. Kawanabe, R. Tomioka, F. Hohlefeld, K.-r. Müller, and V. V. Nikulin. Invariant common spatial patterns: Alleviating nonstationarities in brain-computer interfacing. In *Advances in neural information processing systems*, pages 113–120, 2007.
- [78] W. Samek, C. Vidaurre, K.-R. Müller, and M. Kawanabe. Stationary common spatial patterns for brain-computer interfacing. *Journal of Neural Engineering*, 9(2):026013, 2012.
- [79] A. Bamdadian, C. Guan, K. K. Ang, and J. Xu. Online semi-supervised learning with KL distance weighting for motor imagery-based BCI. In *International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2732–2735. IEEE, 2012.
- [80] M. Arvaneh, C. Guan, K. K. Ang, and C. Quek. EEG data space adaptation to reduce intersession nonstationarity in brain-computer interface. *Neural Computation*, 25(8):2146–2171, 2013.

- [81] M. Krauledat, M. Tangermann, B. Blankertz, and K.-R. Müller. Towards zero training for brain-computer interfacing. *PloS one*, 3(8):e2967, 2008.
- [82] S. Fazli, M. Danóczy, J. Schelldorfer, and K.-R. Müller. 1-penalized linear mixed-effects models for high dimensional data with application to BCI. *NeuroImage*, 56(4):2100–2108, 2011.
- [83] C. Vidaurre, M. Kawanabe, P. Von Bunau, B. Blankertz, and K.-R. Muller. Toward unsupervised adaptation of LDA for brain–computer interfaces. *IEEE Trans. Biomedical Engineering*, 58(3):587–597, 2011.
- [84] M. Sugiyama, M. Krauledat, and K.-R. Müller. Covariate shift adaptation by importance weighted cross validation. *The Journal of Machine Learning Research*, 8:985–1005, 2007.
- [85] A. Satti, C. Guan, D. Coyle, and G. Prasad. A covariate shift minimisation method to alleviate non-stationarity effects for an adaptive brain-computer interface. In *Intl. Conf. on Pattern Recognition*, pages 105–108. IEEE, 2010.
- [86] D. J. Krusienski, M. Grosse-Wentrup, F. Galán, D. Coyle, K. J. Miller, E. Forney, and C. W. Anderson. Critical issues in state-of-the-art brain–computer interface signal processing. *Journal of Neural Engineering*, 8(2):025002, 2011.
- [87] O. Ledoit and M. Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of multivariate analysis*, 88(2):365–411, 2004.
- [88] F. Lotte and C. Guan. Learning from other subjects helps reducing brain-computer interface calibration time. In *IEEE Intl. Conf. on Acoustics Speech and Signal Processing*, pages 614–617. IEEE, 2010.
- [89] A. N. Tikhonov. Regularization of incorrectly posed problems. In *Soviet Math. Dokl*, volume 4, pages 1624–1627, 1963.
- [90] F. Lotte and C. Guan. Spatially regularized common spatial patterns for eeg classification. In *International Conference on Pattern Recognition (ICPR)*, pages 3712–3715. IEEE, 2010.

- [91] M. Arvaneh, G. Cuntai, K. A. Kai, and Q. Chai. Optimizing spatial filters by minimizing within-class dissimilarities in electroencephalogram-based brain-computer interface. *IEEE Trans. Neural Networks and Learning Systems*, 24(4):610–619, 2013.
- [92] K. K. Ang, Z. Y. Chin, H. Zhang, and C. Guan. Filter bank common spatial pattern (FBCSP) in brain-computer interface. In *IEEE International Joint Conference on Neural Networks.(IEEE World Congress on Computational Intelligence).*, pages 2390–2397. IEEE, 2008.
- [93] G. Dornhege, B. Blankertz, M. Krauledat, F. Losch, G. Curio, and K.-R. Müller. Combined optimization of spatial and temporal filters for improving brain-computer interfacing. *IEEE Trans. Biomedical Engineering*, 53(11):2274–2281, 2006.
- [94] R. Tomioka, G. Dornhege, G. Nolte, B. Blankertz, K. Aihara, and K.-R. Müller. Spectrally weighted common spatial pattern algorithm for single trial EEG classification. *Dept. Math. Eng., Univ. Tokyo, Tokyo, Japan, Tech. Rep*, 40, 2006.
- [95] W. Wu, X. Gao, B. Hong, and S. Gao. Classifying single-trial EEG during motor imagery by iterative spatio-spectral patterns learning (ISSPL). *IEEE Trans. Biomedical Engineering*, 55(6):1733–1743, 2008.
- [96] Q. Novi, C. Guan, T. Dat, and P. Xue. Sub-band common spatial pattern (SBCSP) for brain-computer interface. In *Intl. Conf. on Neural Engineering*, pages 204–207, May 2007.
- [97] K. P. Thomas, C. Guan, C. T. Lau, A. P. Vinod, and K. K. Ang. A new discriminative common spatial pattern method for motor imagery brain-computer interfaces. *IEEE Trans. Biomedical Engineering*, 56(11):2730–2733, 2009.
- [98] H. Zhang, Z. Y. Chin, K. K. Ang, C. Guan, and C. Wang. Optimum spatio-spectral filtering network for brain-computer interface. *IEEE Trans. Neural Networks*, 22(1):52–63, 2011.

- [99] J. Meng, L. Yao, X. Sheng, D. Zhang, and X. Zhu. Simultaneously optimizing spatial spectral features based on mutual information for EEG classification. *IEEE Trans. Biomedical Engineering*, 62(1):227–240, 2015.
- [100] H.-I. Suk and S.-W. Lee. A novel bayesian framework for discriminative feature extraction in brain-computer interfaces. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 35(2):286–299, 2013.
- [101] K. P. Thomas, C. Guan, C. T. Lau, A. Vinod, and K. K. Ang. Adaptive tracking of discriminative frequency components in electroencephalograms for a robust brain–computer interface. *Journal of neural engineering*, 8(3):036007, 2011.
- [102] W. Samek, D. Blythe, K.-R. Müller, and M. Kawanabe. Robust spatial filtering with beta divergence. In *Advances in Neural Information Processing Systems*, pages 1007–1015, 2013.
- [103] M. Kawanabe, W. Samek, K.-R. Müller, and C. Vidaurre. Robust common spatial filters with a maxmin approach. *Neural Computation*, 26(2):349–376, 2014.
- [104] W. Samek and M. Kawanabe. Robust common spatial patterns by minimum divergence covariance estimator. In *Intl. Conf. on Acoustics, Speech and Signal Processing*, pages 2040–2043. IEEE, 2014.
- [105] T. N. Lal, M. Schroder, T. Hinterberger, J. Weston, M. Bogdan, N. Birbaumer, and B. Scholkopf. Support vector channel selection in BCI. *IEEE Trans. Biomedical Engineering*, 51(6):1003–1010, 2004.
- [106] M. Schröder, T. N. Lal, T. Hinterberger, M. Bogdan, N. J. Hill, N. Birbaumer, W. Rosenstiel, and B. Schölkopf. Robust EEG channel selection across subjects for brain-computer interfaces. *EURASIP Journal on Applied Signal Processing*, 2005: 3103–3112, 2005.
- [107] J. Yang, H. Singh, E. L. Hines, F. Schlaghecken, D. D. Ilescu, M. S. Leeson, and N. G. Stocks. Channel selection and classification of electroencephalogram signals: An artificial neural network and genetic algorithm-based approach. *Artificial intelligence in medicine*, 55(2):117–126, 2012.

- [108] Y. Wang, S. Gao, and X. Gao. Common spatial pattern method for channel selection in motor imagery based brain-computer interface. In *Intl. Conf. of the Engineering in Medicine and Biology Society*, pages 5392–5395. IEEE, 2006.
- [109] J. Meng, G. Liu, G. Huang, and X. Zhu. Automated selecting subset of channels based on CSP in motor imagery brain-computer interface system. In *Intl. Conf. on Robotics and Biomimetics (ROBIO)*, pages 2290–2294. IEEE, 2009.
- [110] M. Arvaneh, C. Guan, K. K. Ang, and C. Quek. Optimizing the channel selection and classification accuracy in EEG-based BCI. *IEEE Trans. Biomedical Engineering*, 58(6):1865–1873, 2011.
- [111] H. Vikram Shenoy and A. Vinod. An iterative optimization technique for robust channel selection in motor imagery based brain computer interface. In *Intl. Conf. on Systems, Man and Cybernetics (SMC)*, pages 1858–1863. IEEE, 2014.
- [112] F. Goksu, N. F. Ince, and A. H. Tewfik. Sparse common spatial patterns in brain computer interface applications. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 533–536. IEEE, 2011.
- [113] M. L. Marx and R. J. Larsen. *Introduction to mathematical statistics and its applications*. Pearson/Prentice Hall, 2006.
- [114] J. Cohen. *Statistical power analysis for the behavioral sciences*. Academic press, 2013.
- [115] A. K. Das and S. Suresh. An effect-size based channel selection algorithm for mental task classification in brain computer interface. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3140–3145. IEEE, 2015.
- [116] M. Naeem, C. Brunner, R. Leeb, B. Graimann, and G. Pfurtscheller. Seperability of four-class motor imagery data using independent components analysis. *Journal of neural engineering*, 3(3):208, 2006.
- [117] G. Dornhege, B. Blankertz, G. Curio, and K.-R. Müller. Boosting bit rates in noninvasive EEG single-trial classifications by feature combination and multiclass paradigms. *IEEE Trans. Biomedical Engineering*, 51(6):993–1002, 2004.

- [118] G. McLachlan. *Discriminant analysis and statistical pattern recognition*, volume 544. John Wiley & Sons, 2004.
- [119] C.-F. Juang, R. Huang, and W. Cheng. An interval type-2 fuzzy-neural network with support-vector regression for noisy regression problems. *IEEE Trans. Fuzzy Systems*, 18(4):686 – 699, 2010.
- [120] E. Kayacan and M. A. Khanesar. *Fuzzy Neural Networks for Real Time Control Applications: Concepts, Modeling and Algorithms for Fast Learning*. Butterworth-Heinemann, 2015.
- [121] L. A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning. *Information Sciences*, 8(3):199 – 249, 1975.
- [122] J. M. Mendel and R. B. John. Type-2 fuzzy sets made simple. *IEEE Trans. Fuzzy Systems*, 10(2):117–127, 2002.
- [123] Q. Liang and J. Mendel. Interval type-2 fuzzy logic systems: Theory and design. *IEEE Trans. Fuzzy Systems*, 8(5):535 – 550, 2000.
- [124] C. Lee, J. Hong, Y. Lin, and W. Lai. Type-2 fuzzy neural network systems and learning. *Intl. Journal of Computational Cognition*, 1(4):79 – 90, 2003.
- [125] C. Wang, C. Cheng, and T. Lee. Dynamical optimal training for interval type-2 fuzzy neural network (T2FNN). *IEEE Trans. System, Man, and Cybern., Part B: Cybern.*, 34(3):1462 – 1477, 2004.
- [126] H. Hagrass. Comments on dynamical optimal training for interval type-2 fuzzy neural network (T2FNN). *IEEE Trans. System, Man, and Cybern., Part B: Cybern.*, 36(5):1206 – 1209, 2006.
- [127] C. Lee, T. Hu, C. Lee, and Y. Lee. A recurrent interval type-2 fuzzy neural network with asymmetric membership functions for nonlinear system identification. In *Intl. Conf. on Fuzzy Systems*, pages 1496 – 1502. IEEE, 2008.

- [128] K. Park, S. Oh, and W. Pedrycz. Design of interval type-2 fuzzy neural networks and their optimization using real-coded genetic algorithms. In *Intl. Conf. on Fuzzy Systems*, pages 2013 – 2018. IEEE, 2009.
- [129] J. Castro, O. Castillo, P. Melin, and A. Rodriguez-Diaz. A hybrid learning algorithm for a class of interval type-2 fuzzy neural networks. *Information Sciences*, 179(13):2175 – 2193, 2009.
- [130] Z. Deng, K.-S. Choi, L. Cao, and S. Wang. T2FELA: Type-2 fuzzy extreme learning algorithm for fast training of interval type-2 fuzzy logic system. *IEEE Trans. Neural Networks and Learning Systems*, 25(4):664–676, 2014.
- [131] E. Lughofer. *Evolving fuzzy systems-methodologies, advanced concepts and applications*, volume 53. Springer, 2011.
- [132] E. Lughofer, C. Cernuda, S. Kindermann, and M. Pratama. Generalized smart evolving fuzzy systems. *Evolving Systems*, pages 1–24, 2015. doi: 10.1007/s12530-015-9132-6.
- [133] M. Pratama, S. G. Anavatti, M. Joo, and E. Lughofer. pClass: An effective classifier for streaming examples. *IEEE Trans. Fuzzy Systems*, 23(2):369–386, 2015.
- [134] C.-F. Juang and C. Hsu. Reinforcement interval type-2 fuzzy controller design by online rule generation and Q-value-aided ant colony optimization. *IEEE Trans. Systems, Man, and Cybern., Part B: Cybern.*, 39(6):1528 – 1542, 2009.
- [135] C.-F. Juang, C.-H. Hsu, and C.-F. Chuang. Reinforcement self-organizing interval type-2 fuzzy system with ant colony optimization. In *Intl. Conf. on Systems, Man and Cybernetics.*, pages 771–776. IEEE, 2009.
- [136] C.-F. Juang and P. Wang. An interval type-2 neural fuzzy classifier learned through soft margin minimization and its human posture classification application. *IEEE Trans. Fuzzy Systems*, 23(5):1474 – 1487, 2015.
- [137] C.-F. Juang and Y. Tsao. A self-evolving interval type-2 fuzzy neural network with online structure and parameter learning. *IEEE Trans. Fuzzy Systems*, 16(6):1411 – 1424, 2008.

- [138] C.-F. Juang and Y. Tsao. A type-2 self-organizing neural fuzzy system and its FPGA implementation. *IEEE Trans. Systems, Man and Cybern.*, 38(6):1537 – 1548, 2008.
- [139] C.-F. Juang, R. Huang, and Y. Lin. A recurrent self-evolving interval type-2 fuzzy neural network for dynamic system processing. *IEEE Trans. Fuzzy Systems*, 17(5): 1092–1105, 2009.
- [140] J. Chang, Y. Lin, M. Han, and C. Lin. A functional link based interval type-2 compensatory fuzzy neural network for nonlinear system modeling. In *IEEE Intl. on Conf. Fuzzy Systems*, pages 939 – 943, 2011.
- [141] Y. Lin, S. Liao, J. Chang, and C. Lin. Simplified interval type-2 fuzzy neural networks. *IEEE Trans. Neural Networks and Learning Systems*, 25(5):959 – 969, 2014.
- [142] G. Wu and P. Huang. A vectorization-optimization-method-based type-2 fuzzy neural network for noisy data classification. *IEEE Trans. Fuzzy Systems*, 21(1):1 – 15, 2013.
- [143] S. Tung, C. Quek, and C. Guan. eT2FIS: An Evolving Type-2 Neural Fuzzy Inference System. *Information Sciences*, 220(1):124 – 148, 2013.
- [144] P. Herman, G. Prasad, and T. M. McGinnity. Investigation of the type-2 fuzzy logic approach to classification in an EEG-based brain-computer interface. In *Intl. Conf. of the Engineering in Medicine and Biology Society*, pages 5354–5357. IEEE, 2005.
- [145] P. Herman, G. Prasad, and T. M. McGinnity. Design and on-line evaluation of type-2 fuzzy logic system-based framework for handling uncertainties in BCI classification. In *Intl. Conf. of the IEEE Engineering in Medicine and Biology Society*, pages 4242–4245. IEEE, 2008.
- [146] P. Herman, G. Prasad, and T. M. McGinnity. Designing a robust type-2 fuzzy logic classifier for non-stationary systems with application in brain-computer interfacing. In *IEEE Intl. Conf. on Systems, Man and Cybern.*, pages 1343–1349. IEEE, 2008.

- [147] T. Nguyen, A. Khosravi, D. Creighton, and S. Nahavandi. EEG signal classification for BCI applications by wavelets and interval type-2 fuzzy logic systems. *Expert Systems with Applications*, 42(9):4370–4380, 2015.
- [148] E. Lughofer. Single-pass active learning with conflict and ignorance. *Evolving Systems*, 3(4):251–271, 2012.
- [149] K. Subramanian, A. K. Das, S. Suresh, and R. Savitha. A meta-cognitive interval type-2 fuzzy inference system and its projection based learning algorithm. *Evolving Systems*, 5(4):219–230, 2014.
- [150] G. S. Babu and S. Suresh. Parkinson’s disease prediction using gene expression - a projection based learning meta-cognitive neural classifier approach. *Expert Systems with Applications*, 40(5):1519 – 1529, 2013.
- [151] K. Subramanian and S. Suresh. A meta-cognitive sequential learning algorithm for neuro-fuzzy inference system. *Applied Soft Computing*, 12(11):3603 – 3614, 2012.
- [152] K. Subramanian and S. Suresh. Human action recognition using meta-cognitive neuro-fuzzy inference system. *International Journal of Neural Systems*, 22(06):1250028, 2012.
- [153] K. Subramanian, S. Suresh, and N. Sundararajan. A meta-cognitive neuro-fuzzy inference system (mcfis) for sequential classification problems. *IEEE Trans. Fuzzy Systems*, 21(6):1080 – 1095, 2013.
- [154] A. K. Das, K. Subramanian, and S. Suresh. An evolving interval type-2 neuro-fuzzy inference system and its meta-cognitive sequential learning algorithm. *IEEE Trans. Fuzzy Systems*, 23(6):2080–2093, 2015.
- [155] A. K. Das, S. Suresh, N. Sundararajan, and K. Subramanian. A subject-specific frequency band selection for efficient BCI- an interval type-2 fuzzy inference system approach. In *IEEE Intl. Conf. on Fuzzy Systems.*, pages 1–8. IEEE, 2015.

- [156] A. K. Das, S. Suresh, and N. Sundararajan. A self-regulated interval type-2 neuro-fuzzy inference system for handling non-stationarities in EEG signals for BCI. *IEEE Trans. Fuzzy Systems*, PP(99):1–1, 2016. ISSN 1063-6706. doi: 10.1109/TFUZZ.2016.2540072.
- [157] P. Angelov, E. Lughofer, and X. Zhou. Evolving fuzzy classifiers using different model architectures. *Fuzzy Sets and Systems*, 159(23):3160–3182, 2008.
- [158] H.-J. Rong, G.-B. Huang, N. Sundararajan, and P. Saratchandran. Online sequential fuzzy extreme learning machine for function approximation and classification problems. *IEEE Trans. Systems, Man, and Cybern., Part B: Cybern.*, 39(4):1067–1072, 2009.
- [159] H.-J. Rong, N. Sundararajan, G.-B. Huang, and P. Saratchandran. Sequential adaptive fuzzy inference system (SAFIS) for nonlinear system identification and prediction. *Fuzzy Sets and Systems*, 157(9):1260–1275, 2006.
- [160] K. Subramanian, S. Suresh, and R. Venkatesh Babu. Meta-cognitive neuro-fuzzy inference system for human emotion recognition. In *Intl. Joint Conf. on Neural Net.*, pages 1 – 7, 2012.
- [161] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:328 – 353, 1965.
- [162] N. Karnik, J. Mendel, and Q. Liang. Type-2 fuzzy logic systems. *IEEE Trans. on Fuzzy Systems*, 7(6):643 – 658, 1999.
- [163] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. Systems, Man and Cybernetics*, SMC-15(1): 116 – 132, 1985.
- [164] M. E.H. Application of fuzzy algorithms for control of simple dynamic plant. In *Proceedings of the Institution of Electrical Engineers*, 220:1585–1588, 1974.
- [165] C.-F. Juang, R.-B. Huang, and W.-Y. Cheng. An interval type-2 neural fuzzy chip with on-chip incremental learning ability for time-varying data sequence prediction and system control. *IEEE Trans. Neural Networks and Learning Systems*, 25(1): 216–228, 2014.

- [166] C. Lee, Y. Lin, and W. Lai. Systems identification using type-2 fuzzy neural network (type-2 FNN) systems. In *Intl. Symposium on Comp. Intel. in Robotics and Automation*, volume 3, pages 1264 – 1269, 2003.
- [167] A. Cai, C. Quek, and D. L. Maskell. Type-2 GA-TSK fuzzy neural network. In *IEEE Congress on Evolutionary Computation*, pages 1578–1585. IEEE, 2007.
- [168] R. A. Aliev., C. Pedrycz., B. G. Guirimov., R. R. Aliev., U. Ilhan, M. Babagil, and S. Mammadli. Type-2 fuzzy neural networks with fuzzy clustering and differential evolution optimization. *Information Sciences*, 181:1591 – 1608, 2011.
- [169] R. H. Abiyev and O. Kaynak. Type-2 fuzzy neural structure for identification and control of time-varying plants. *IEEE Trans. Industrial Electronics*, 57(12): 4147–4159, 2010.
- [170] C.-F. Juang and C.-Y. Chen. Data-driven interval type-2 neural fuzzy system with high learning accuracy and improved model interpretability. *IEEE Trans. Cybernetics*, 43(6):1781–1795, 2013.
- [171] M. A. Khanesar, E. Kayacan, M. Teshnehlav, and O. Kaynak. Extended Kalman filter based learning algorithm for type-2 fuzzy logic systems and its experimental evaluation. *IEEE Trans. Industrial Electronics*, 59(11):4443–4455, 2012.
- [172] O. Castillo and P. Melin. Optimization of type-2 fuzzy systems based on bio-inspired methods: A concise review. *Information Sciences*, 205:1 – 19, 2012.
- [173] M. Sugeno and T. Yasukawa. A fuzzy-logic-based approach to qualitative modeling. *IEEE Trans. Fuzzy Systems*, 1(1):7–31, 1993.
- [174] C.-F. Juang and K. Juang. Reduced interval type-2 neural fuzzy system using weighted bound-set boundary operation for computation speedup and chip implementation. *IEEE Trans. Fuzzy Systems*, 21(3):477–491, 2013.
- [175] M. Pratama, J. Lu, and G. Zhang. Evolving type-2 fuzzy classifier. *IEEE Trans. Fuzzy Systems*, 24(3):574–589, 2016.

- [176] Y. Lin, J. Chang, N. Pal, and C. Lin. A mutually recurrent interval type-2 neural fuzzy system (MRIT2NFS) with self-evolving structure and parameters. *IEEE Trans. Fuzzy Systems*, 21(3):492 – 509, 2013.
- [177] Y.-Y. Lin, J.-Y. Chang, and C.-T. Lin. A tsf-type-based self-evolving compensatory interval type-2 fuzzy neural network (TSCIT2FNN) and its applications. *IEEE Trans. Industrial Electronics*, 61(1):447–459, 2014.
- [178] J. R. Castro, O. Castillo, P. Melin, and A. Rodríguez-Díaz. A hybrid learning algorithm for interval type-2 fuzzy neural networks: the case of time series prediction. In *Soft Computing for Hybrid Intelligent Systems*, pages 363–386. Springer, 2008.
- [179] M. Pratama, J. Lu, E. Lughofer, G. Zhang, and S. Anavatti. Scaffolding type-2 classifier for incremental learning under concept drifts. *Neurocomputing*, 191: 304–329, 2016.
- [180] R. M. Isaacson and F. Fujita. Metacognitive knowledge monitoring and self-regulated learning: Academic success and reflections on learning. *Journal of Scholarship of Teaching and Learning*, 6(1):39–55, 2006.
- [181] J. H. Flavell. Metacognition and cognitive monitoring: A new area of cognitive–developmental inquiry. *American psychologist*, 34(10):906, 1979.
- [182] M. T. Cox. Field review: Metacognition in computation: A selected research review. *Artificial intelligence*, 169(2):104–141, 2005.
- [183] K. Subramanian, R. Savitha, and S. Suresh. A complex-valued neuro-fuzzy inference system and its learning mechanism. *Neurocomputing*, 123(0):110 – 120, 2014.
- [184] G. S. Babu and S. Suresh. Meta-cognitive RBF network and its projection based learning algorithm for classification problems. *Applied Soft Computing*, 13(1):654 – 666, 2013.
- [185] G. S. Babu and S. Suresh. Meta-cognitive neural network for classification problems in a sequential learning framework. *Neurocomputing*, 81(1):86 – 96, 2012.

- [186] R. Savitha, S. Suresh, and N. Sundararajan. Metacognitive learning in a fully complex valued radial basis function neural network. *Neural Computation*, 24(5): 1297 – 1328, 2012.
- [187] R. Savitha, S. Suresh, and N. Sundararajan. A meta-cognitive learning algorithm for a fully complex-valued relaxation network. *Neural Networks*, 32(Special Issue): 209 – 218, 2012.
- [188] S. Suresh, R. Savitha, and N. Sundararajan. A sequential learning algorithm for complex valued self regulating resource allocation network- CSRAN. *IEEE Trans. Neural Networks*, 22(7):1061 – 1072, 2011.
- [189] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.
- [190] W. H. Press. *Numerical recipes 3rd edition: the art of scientific computing*. Cambridge university press, 2007.
- [191] C. Blake and C. Merz. UCI repository of machine learning databases, 1998. URL <http://archive.ics.uci.edu/ml/>. Department of Information and Computer Sciences, University of California, Irvine.
- [192] A. Schlögl, F. Lee, H. Bischof, and G. Pfurtscheller. Characterization of four-class motor imagery EEG data for the BCI-competition 2005. *Journal of Neural Engineering*, 2(4):L14, 2005.
- [193] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.
- [194] 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*, 6(39):1–9, 2012.
- [195] A. S. Royer, A. J. Doud, M. L. Rose, and B. He. EEG control of a virtual helicopter in three-dimensional space using intelligent control strategies. *IEEE Trans. Neural Systems and Rehabilitation Engineering*, 18(6):581–589, 2010.

- [196] A. J. Doud, J. P. Lucas, M. T. Pisansky, and B. He. Continuous three-dimensional control of a virtual helicopter using a motor imagery based brain-computer interface. *PloS one*, 6(10):e26322, 2011.
- [197] K. LaFleur, K. Cassady, A. Doud, K. Shades, E. Rogin, and B. He. Quadcopter control in three-dimensional space using a noninvasive motor imagery-based brain-computer interface. *Journal of neural engineering*, 10(4):046003, 2013.
- [198] B. H. Kim, M. Kim, and S. Jo. Quadcopter flight control using a low-cost hybrid interface with EEG-based classification and eye tracking. *Computers in biology and medicine*, 51:82–92, 2014.
- [199] J.-S. Lin and Z.-Y. Jiang. Implementing remote presence using quadcopter control by a non-invasive BCI device. *Computer Science and Information Technology*, 3(4):122–126, 2015.
- [200] EMOTIV. <http://emotiv.com/>. Accessed: 2016-05-05.
- [201] H. H. Jasper. The ten twenty electrode system of the international federation. *Electroencephalography and clinical neurophysiology*, 10:371–375, 1958.
- [202] AR.Drone2.0. <http://www.parrot.com/usa/products/ardrone-2/>. Accessed: 2016-05-05.
- [203] K. Subramanian and S. Savitha, R. and Suresh. Zero-error density maximization based learning algorithm for a neuro-fuzzy inference system. In *Intl. Conf. on Fuzzy Systems*, pages 1–7. IEEE, 2013.
- [204] F. Fernández-Navarro, C. Hervás-Martínez, P. A. Gutiérrez, J. M. Peña-Barragán, and F. López-Granados. Parameter estimation of q-gaussian radial basis functions neural networks with a hybrid algorithm for binary classification. *Neurocomputing*, 75(1):123–134, 2012.
- [205] J. M. Mendel. *Uncertain rule-based fuzzy logic systems: introduction and new directions*. 2001.