Evolutionary Optimization Approach For Optimal EEG Feature Selection For Effective Cognitive Task Classification

By

Mousumi Roy

Registration No.:137294 of 2016-2017

Examination Roll No.: M6IAR19004

Under the guidance of

Dr. Pratyusha Rakshit

This thesis is submitted in the partial fulfillment for the Degree of Master of Technology in Intelligent Automation and Robotics under Electronics and Telecommunication Engineering

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

Jadavpur University

Kolkata-700032

May 2019

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

CERTIFICATE OF RECOMMENDATION

This is to certify that the thesis entitled "**Evolutionary Optimization Approach For Optimal EEG Feature Selection For Effective Cognitive Task Classification**" has been carried out by Mousumi Roy (University Registration No.: 137294 OF 2016-2017) under my guidance and supervision and be accepted in partial fulfillment of the requirement for the degree of Master of Technology in Intelligent Automation and Robotics of Jadavpur University.

Dr. Pratyusha Rakshit
(Project Guide)
Intelligent Automation and Robotics
Department of Electronics &
Telecommunication Engineering
Jadavpur University

Prof. Amit Konar
(Course Coordinator)
Intelligent Automation and Robotics
Department of Electronics &
Telecommunication Engineering
Jadavpur University

Dr. Sheli Sinha Chaudhuri
(Head of the Department)
Department Of Electronics & Telecommunication Engineering
Jadavpur University

Prof. Chiranjib Bhattacharjee
Dean, Faculty Council Of Engineering
and Technology
Jadavpur University

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

CERTIFICATE OF APPROVAL

The foregoing thesis is hereby approved as a creditable study of engineering subject and presented in a manner satisfactory to warrant acceptance as prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn there in but approve the thesis only for which it is submitted.

Mousumi Roy

Examination Roll No.: M6IAR19004

Committee on final examination for evaluation of the thesis

External Examiner

Dr. Pratyusha Rakshit
(Supervisor)

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

DECLARATION OF ORIGINALITY OF COMPLIANCE
OF
ACADEMIC THESIS

I hereby declare that the thesis entitled "**Evolutionary Optimization Approach For Optimal EEG Feature Selection For Effective Cognitive Task Classification**" contains literature survey and original research work by the undersigned candidate, as part of her Degree of Master of Technology in Intelligent Automation and Robotics.

All information have been obtained and presented in accordance with the academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials when required and none of the work represented in this thesis is fabricated.

Name: Mousumi Roy

Examination Roll No.: M6IAR19004

Thesis Title: Evolutionary Optimization Approach For Optimal EEG Feature Selection For effective Cognitive Task Classification

Date:

Place: Kolkata

Signature of candidate

ACKNOWLEDGEMENTS

First and foremost, I would like to express my earnest gratitude and heartfelt indebtedness to my advisor , Dr. Pratyusha Rakshit, Department of Electronics and Telecommunication Engineering, for the privilege of allowing me to work under her towards my Degree Of Master Of Electronics & Telecommunication Engineering. This work would not have materialized, but for her whole-hearted help and support. Working under her has been a great experience. I sincerely thank my supervisor, particularly for all faith she had in me. I am thankful to Dr. Sheli Sinha Chaudhuri who acted as Head of the Department of Electronics and Telecommunication Engineering during the tenure of my studentship. I would like to show my gratitude to the respected professors of the Department of Electronics and Telecommunication Engineering particularly Prof Amit Konar for his constant guidance and valuable advices.

I am grateful to my seniors for their constant support and encouragement. I am indebted to my classmates and friends for the good wishes they have provided. Lastly, I would like to thank my family for their love and support.

Date:

Place: Kolkata

Mousumi Roy
Examination Roll No.: M6IAR19004
Jadavpur University

Table of Contents

Table of Contents

List of Figures:

List of Charts:

Chapter 1: Overview of Brain Computer Interfacing

1.1: Brain Computer Interfacing:

A brain–computer interface (abbreviated as BCI), also known as neural-control interface (NCI), mind-machine interface (MMI), direct neural interface (DNI) or brain–machine interface (BMI), is a technology that facilitates direct, real-time communication between a human or an animal brain and an external device. The intent of the user is identified from his brain signal which then translated into commands to accomplish the desire of the user.

The main aim of BCI is to replace normal brain's neuro-muscular output pathways and instead use brain signals to carry out specific task. The process involves acquisition of brain signals, analyzing them and then translating them into commands which are later fed into external devices to perform desired task.

For example: BCI enables a paralytic person to write a book or control a motorized wheelchair or prosthetic limb through thought alone by reading signals from an array of neurons in brain and using computer chips and programs to translate the signals into action.

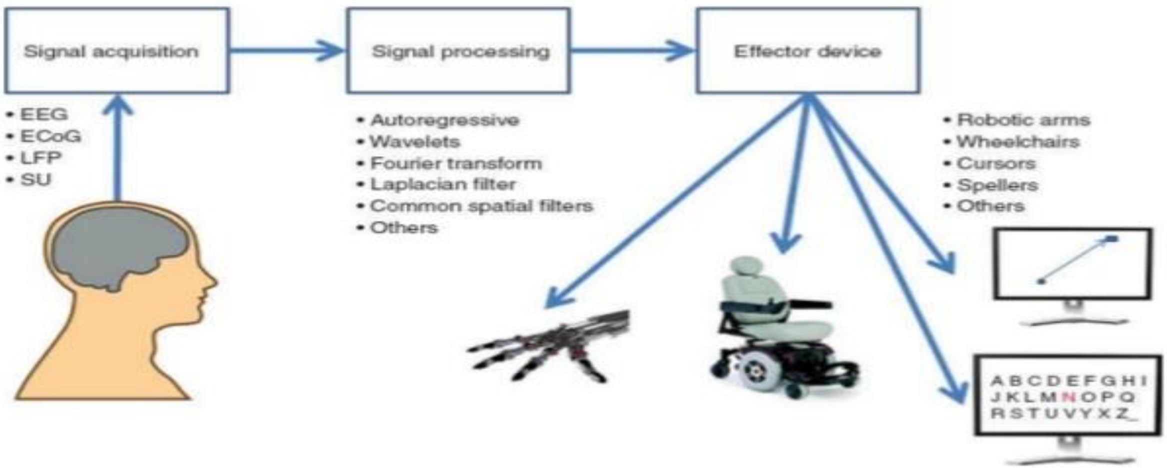


Fig 1.1: A general BCI system

How it started? A look at background..........

The history of BCI dates back to 1924 when Hans Berger discovered the electrical activity of brain and developed the concept of Electroencephalography (EEG).He was able to identify the oscillatory activity, which later came to known as Alpha Wave (8-13Hz) by analyzing EEG traces.

The term BCI was coined by ULCA Professor Jacques Vidal. He is widely recognized as the inventor of BCIs in the BCI community and was also the first person to produce the first peer-reviewed publications on this topic. In 1977, Vidal demonstrated a noninvasive EEG (Visual Evoked Potentials (VEP)) control of a cursor-like graphical object on a computer screen.

In 1969, Fetz and his colleagues at University of Washington School of Medicine in Seattle, showed for the first time that monkeys could learn to voluntarily control the firing rates of neurons and hence could control biofeedback meter arm with neural activity.

Since mid 1990's several group of researchers were able to capture brain motor cortex signals and used the signals to control external devices.

In 1999, a team of researchers led by Yang Dan (at University of California) were able to produce images seen by cats. An array of electrodes was embedded in the thalamus of sharp-eyed cats. The cats were shown eight short movies and the electrodes integrated the neuronal firings. Then the brain's signal was decoded by some filters to produce images seen by cats.

Experiments conducted by Miguel Nicolelis of Duke University using rhesus monkeys succeeded in closing the feedback loop and reproduced monkey reaching and grasping movements in a robot arm.

The Current Scenario……….

Since the last two decades, there has been a huge progression in the field from mere neuroscientific discovery to initial translational applications. Collaboration of neuro-science with engineering have materialized the concept of BCI to reality. As of now, BCI finds a wide range of application starting from clinical to security, controlling appliances to name a few. Lie detector, cognitive monitoring, tele-presence, gaming, human augmentation are some of the recent trends of BCI. The application of BCI in medical domain is huge and plays a very role significant. Starting from neuroprosthetics to neurorehabilitation, BCI platforms are a boon to the patients whose motor cortex has been traumatized by accidents or diseases.

What's next? The possibilities…….

A type of BCI which is still theoretical is the creation of technologies that are more intelligent than humans. A technology is thought of where the users could upload an entire human brain to a theoretical type of BCI that would completely replicate its function, allowing a human to live on in terms of brain function, without a human body.

Again, it is expected that the existing shortcomings of BCI technologies could be overcome in near future and it could provide higher levels of complexity in brain-derived control and extend its applications to populations such as those with hemispheric stroke or brain trauma.

These and other similar kind of ideas continue to drive interest in developing more efficient and sophisticated BCIs that can more completely receive a model of high-level human brain activity or intelligence and could be of potential benefit to the humans.

1.2: Examples of BCI:

A. BCI in Neuro-prosthetic control:

Neuroprosthetics defined as a device that supplants or supplements the input and/or output of the nervous system. It involves the use of BCI technique to control the movements of limbs or arms in case of paralytic, quadriplegic person or amputees. In the process, the person thinks of a specific direction and his brain activity gets recorded via electrodes placed on scalp near the nerves responsible for specific movements. Based on the direction he chooses, a particular region (Beta wave) of his brain gets modulated, which in turn controls the movement of cursor on screen. This signal from the cursor is fed back to a joystick which controls the movement of his limbs or arms.

Fig 1.2: Person with bionic arm

B. BCI Systems as a means of communication:

Individuals affected with ALS, cerebral palsy, brainstem stroke, spinal cord injuries, muscular dystrophies, or chronic peripheral neuropathies that have little or no neuro-muscular control can utilize the BCI technology to carry out their basic communication. Three types of EEG-based BCI systems are generally used for communication-

1. slow cortical potentials (SCPs)
2. P300 event-related potentials
3. Sensorimotor rhythms (SMRs)

The user learns to control the SCP-BCI and SMR-BCI with severe trainings (over months) whereas P300 requires less severe trainings.P300 are basically brain's response to auditory and visual responses. SCP and SMR s are slow voltage changes in cortex and sensorimotor cortex respectively. The SCP base BCI converts the voltage changes into vertical movements of cursor in computer screen. Several research works have proved that these signals can be used by BCI systems to produce communication functions like spellers, icon selectors, word processors etc.

Fig 1.3: BCI assisted communication

C. <u>BCI assisted Locomotion:</u>

An electric wheel-chair controlled by EEG is an example of BCI assisted locomotion. The user continuously generates commands for the robot that are then probabilistically combined with pre-wired behaviors. The goal of the user is decoded by the sensors (usually laser sensors) fitted with the wheel-chair in the form of the form of a probability distribution over a set of possible mental steering commands. The EEG- based BCI system estimates the probabilities for the different mental commands from the user's brain signals. The wheelchair is controlled using a filtered estimate of the user's intent. The command with the highest probability is used to control the wheelchair.

Fig 1.4: BCI controlled wheel-chair

D. <u>BCI in Neuro-rehabilitation:</u>

BCI also aims to restore the normal motor function to those individuals whose neuromuscular function has been damaged by accidents, trauma or diseases. This approach utilizes brain signals from healthy individuals along with the decoded kinematic parameters. The signals when fed to the patients help them to modify their thinking patterns so that they can resemble the recorded patterns. This causes the patients to produce more normal brain activities, which in turn enables them to more CNS function and enable motor control.

E. <u>BCI in Entertainment & Gaming:</u>

The existing feature of the games are combined with brain controlling capabilities. These technologies basically use the EEG technique to receive signals from the brain and use it for various purposes like moving a cursor on the screen or guiding the movements of an avatar in a virtual environment by imagining these movements . NeuroBoy, Judecca, Mindflex, Star wars Force are some of the examples.

Fig 1.5: BCI controlled gaming

1.3: Different steps of BCI:

The different steps of BCI are discussed briefly below-

A. <u>Signal Acquisition</u>:

The signals are acquired from the scalp of the subject by the use of electrodes. There are different types of electrodes available for use in EEG, such as:

-disposable (dry or wet)

-reusable disc electrodes (gold, silver, stainless steel or tin)

- head-bands and electrodes caps (such as the consumer ones)

-saline based electrodes

-needle electrodes.

The minimal configuration is composed by three electrodes: active electrode, reference electrode and ground electrode. The EEG measures the potential difference over time between signal or active electrode and the reference electrode. Reference is placed on the mastoid, ear lobes or tip of the nose. The ground electrode is used to measure the differential voltage between the active and the reference points. Usually, the International 10-20 system of electrode placement is followed for acquisition of brain signals.

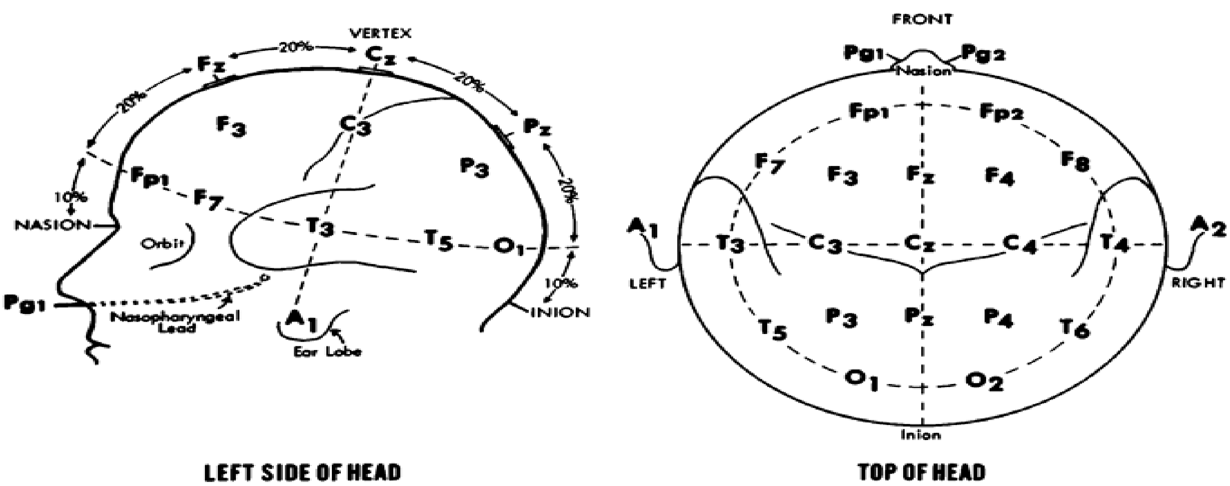


Fig 1.6: International 10-20 system for EEG recording

Amplification

The signal acquired by the electrodes is very weak (usually in the range of micro-volts) and attenuated by the different layers of the brain through which it travels. Hence, amplification of these signals are required to bring it to the range that can be digitized.

A/D conversion

The A/D converter converts the amplified signal from analog to digitalized form. The bandwidth for EEG signals is limited to 100Hz (approx), making 200Hz enough for sampling of EEG signals.

Recording of signal

The signal is then recorded in a computer or similar device, for storing and displaying the converted signal.

Preprocessing

The raw EEG data is often affected by noise and artifacts. There are four main sources of noise and artifacts, which are:

-EEG equipment

-Electrical interference external to the subject and recording system

-The leads and the electrodes

-The subject (the electrical activity from the heart, eye blinking, eyeball movements, muscles movements)

The purpose of the preprocessing step is to clean the data from the noise and artifacts. There are different methods and different steps in preprocessing.

Sometimes, filters are applied to the data to remove the DC components of the signal and the drifts are employed high-pass filters (usually a frequency cut-off of 1Hz is enough). Since in EEG, frequencies above 90 Hz are discarded, sometimes also low pass filters are applied to eliminate the high frequencies of the signal. Some other methods are used to remove artifacts as the eyeball movements or eye blinking.

After different steps of pre-processing, the recording is cut in epoch of few seconds. This allows us to have a large number of features from a single EEG recording, and to use them for statistics or to apply classifiers.

Feature Extraction

Feature extraction helps in the analysis of the signal and extraction of relevant information. Since EEG signal is very complex, it is not possible to find meaningful out information just looking at it. Processing algorithms allows to find content (such as a person's intent, for example) which would be hidden at a naked eye.

There are many methods for feature extraction, some of them are:

-Wavelet transform (WT)

-Independent Component Analysis (ICA)

- Autoregressive Modeling (AR)

-Empirical Mode Decomposition (EMD)

-Principal Component Analysis (PCA)

Classification:

Using machine learning techniques it is possible to train a classifier to recognize which features, for example, belongs to one or another class. Classification helps to find out which kind of mental task the subject is performing.

Translation

After classification of the signal, the result is passed to the feature translation algorithm. Here, features are translated to the corresponding action required. As an example, a P3 potential is translated into the selection of the letter that evoked it. So, in this case, the algorithm will send a command to the feedback device, to select the letter.

Feedback device

Ultimately, the feedback device receives the command from the translation step. For example it could be the computer, where the signal will be used to move a cursor, or it could be a robotic arm where the data are used to allow movement.

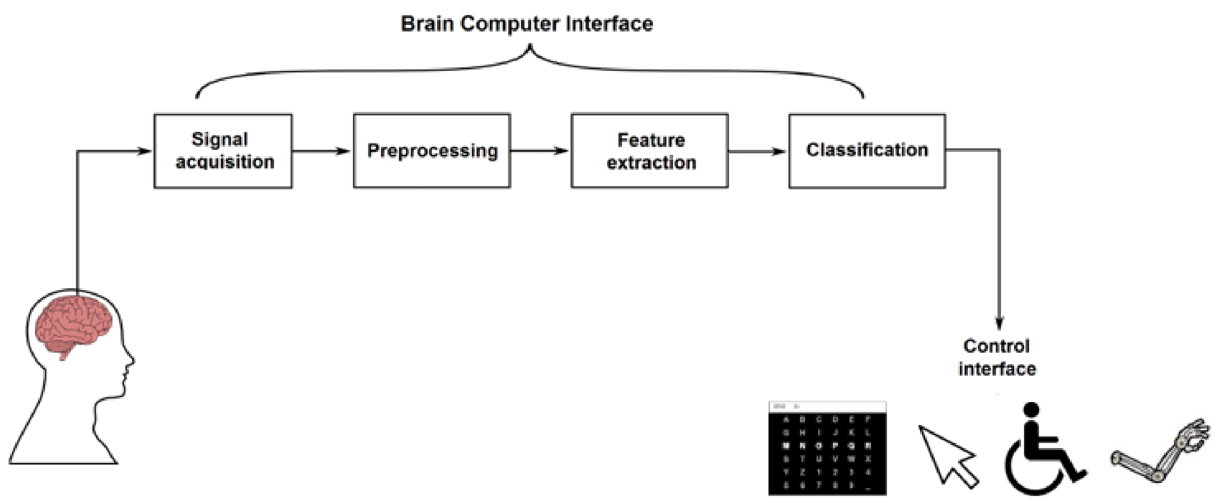


Fig 1.7: Different steps of Brain Computer Interfacing

1.4: Different modalities of Brain Computer Interfacing:

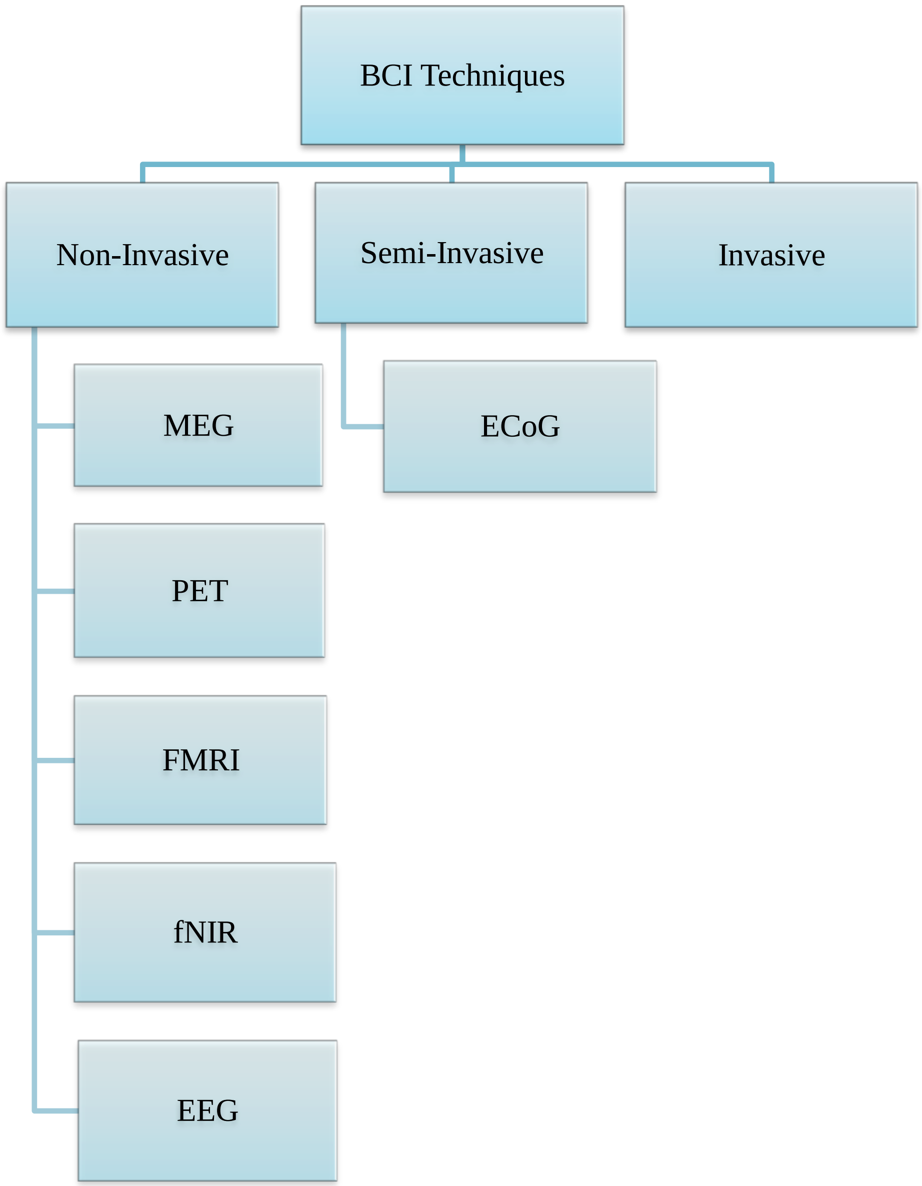


Chart 1.1: Different Modalities of Brain Computer Interfacing

<u>**Non-invasive:**</u> The sensors are placed on the scalp to measure the electrical potentials produced by the brain (EEG) or the magnetic field (MEG).

<u>Semi-invasive:</u> The ECoG signal is taken from electrodes placed in the dura or in the arachnoid.

<u>Invasive:</u> The Intraparenchymal signal is taken directly implanting electrodes in the cortex.

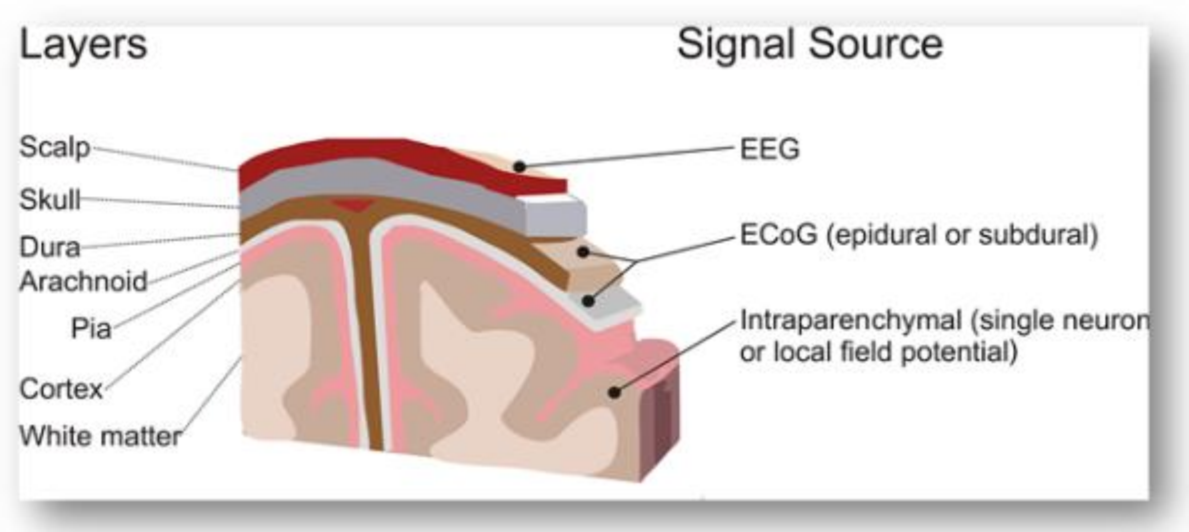


Fig 1.8: Different BCI techniques applied on different layers of brain

The different modalities of BCI are discussed in detail below-

<u>Magneto encephalography (MEG)</u>

It is defined as the functional neuroimaging technique for mapping brain activity by recording magnetic fields produced by electrical currents occurring naturally in the brain, using very sensitive magnetometers. In simple words, it is a noninvasive technique for investigating brain activity and it allows the measurement of ongoing brain activity on a millisecond-by-millisecond basis, and it shows where in the brain activity is produced.

The advantages of using MEG are:

-provides timing as well as spatial information about brain activity.

-signals are able to show absolute neuronal activity.

-can be recorded in sleeping subjects.

-its measurement does not require the complete absence of subject movement during recording.

-provides us with temporal characteristics about brain activation with sub-millisecond precision.

Fig 1.9: Magneto encephalography (MEG)

Positron emission tomography (PET)

It is a nuclear imaging technique used in medicine to observe different processes, such as blood flow, metabolism, neurotransmitters, happening in the body. It can be used to restore sensory and motor function in patients with neurologic disorders.

Fig 1.10: Positron emission tomography (PET)

Functional magnetic resonance imaging (FMRI)

It is the functional neuroimaging technique (using Magnetic Resonance Imaging technology) to measure activity of brain by detecting changes associated with blood flow. This technique relies on the fact that cerebral blood flow and neuronal activation are coupled. When an area of the brain is in use, blood flow to that region gets increased.

Fig 1.11: functional magnetic resonance imaging (FMRI)

Functional Near-infrared spectroscopy (fNIRS)

It is the use of near-infrared spectroscopy for the purpose of functional neuroimaging. Brain activity is measured through hemodynamic responses associated with neuron behavior. It measures the oxygen concentration levels from the brain, similar to functional magnetic resonance imaging (fMRI). Compared to fMRI, fNIRS has a much higher temporal resolution which allows measurements of concentration changes in both oxygenated and deoxygenated hemoglobin. The advantages of using fNIRS are its non-invasiveness, portability and low-cost method of indirect and direct monitoring of brain activity.

Fig 1.12: Functional Near-infrared spectroscopy (fNIRS)

Electroencephalography (EEG)

It is an electrophysiological monitoring method to record electrical activity of the brain. It is noninvasive method in with the electrodes placed along the scalp. It measures the fluctuations of voltage that occurs from ionic current within the neurons of the brain. Clinically, it is used for diagnosis of conditions such as seizures, epilepsy, head injuries, dizziness, headaches, brain tumors, sleeping problems and it confirms brain death also.

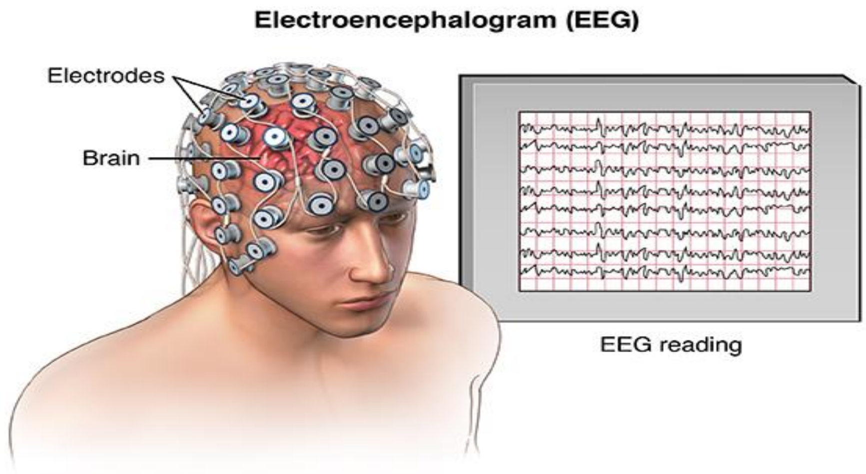


Fig 1.13: Electroencephalography (EEG)

Invasive

In invasive BCI, an electrode array is directly implanted into the brain during a neurosurgery. There are-

- single unit BCIs that detect the signal from a single area of brain
- multi-unit BCIs that detects the signal from multiple areas of brain

Electrodes used in invasive BCI are of different lengths like up to 1.5 mm or 10 mm.

The quality of the signal obtained from this technique is the highest, but the procedure has several issues, like it involves the risk of forming scar tissues. Also, the body develops reactions to the foreign object and builds the scar around the electrodes, which cause deterioration in the signal. The targets of invasive BCI are mainly blind and paralyzed patients as it very risky to operate.

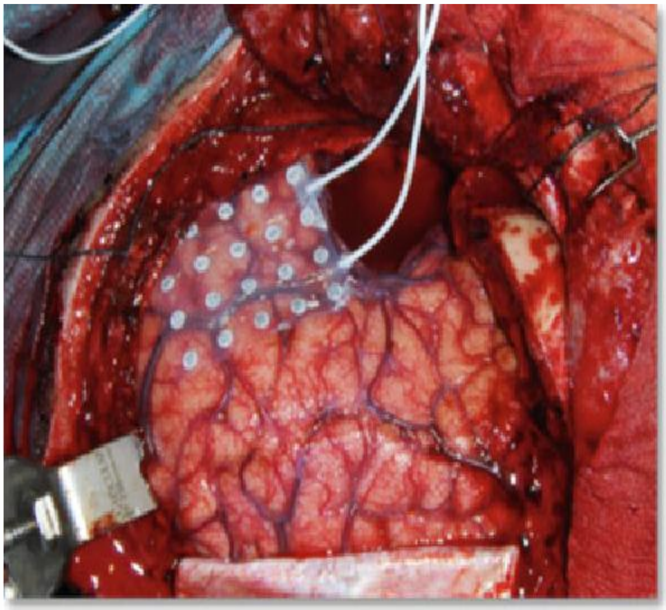


Fig 1.14: Invasive Brain Computer Interfacing

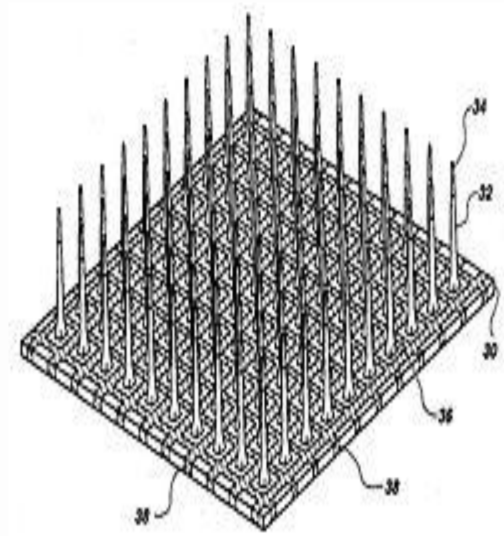


Fig 1.15: Electrode array

<u>Electrocorticography (ECoG)</u>

It involves the uses electrodes placed on the exposed surface of the brain to measure electrical activity from the cerebral cortex. It is called semi-invasive but it still requires a craniotomy to implant the electrodes. For this reason it is used only when surgery is necessary for medical reasons (epilepsy for example).

The electrodes may be placed outside the dura mater (epidural) or under the dura mater (subdural). The strip or grid electrodes cover a large area of the cortex.

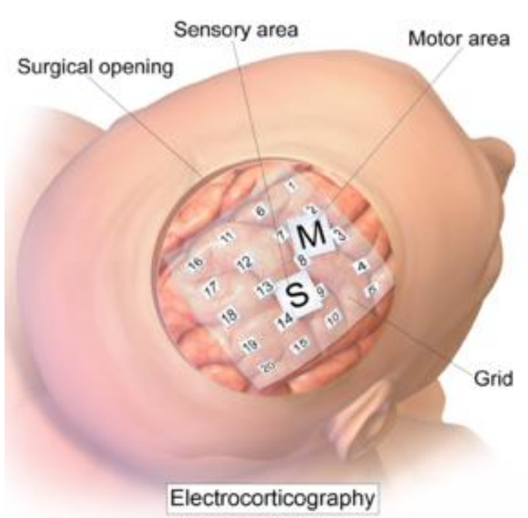


Fig 1.16: Semi-invasive BCI

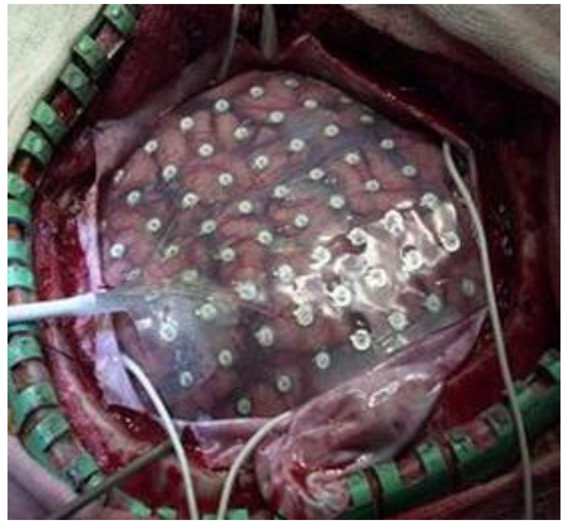


Fig 1.17: Electrocorticography (ECoG)

1.5: Significance of EEG-based BCI:

1. Hardware costs are lower in comparison to other imaging techniques such as MRI scanning.

2. EEG sensors can be deployed into a wide variety of environments.

3. EEG allows higher temporal resolution on the order of milliseconds.

4. EEG is relatively tolerable to subject movements as compared to MRI.

5. The silent nature of EEG allows for better study of responses.

6. EEG can be used in subjects that are not capable of making a motor response.

7. In EEG some voltage components can be detected even when the subject is not responding to stimuli.

1.5: Features of EEG Signal:

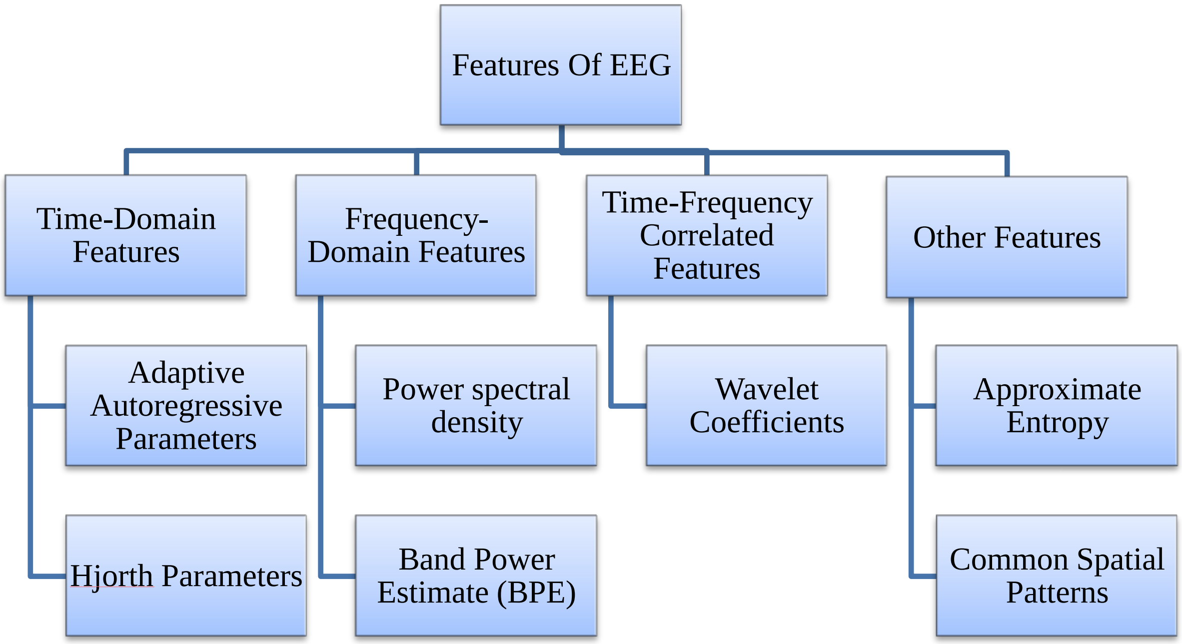


Chart 1.2: Different features of EEG signal

<u>Adaptive autoregressive parameters:</u>

An adaptive autoregressive (AAR) model is used for time-varying spectral analysis of the EEG. It has important role in sleep analysis. It describes the stochastic behavior of a time series. An AAR model is described by the following equation –

$$y_k = a_1 * y_{k-1} + \cdots + a_p * y_{k-p} + x_k$$

with $x_k = N\{0, \sigma_x^2\}$

x_k is a zero-mean-Gaussian-noise process with variance σ_x^2; k is an integer number and describes discrete, equidistant time points; p is the order of the AR model and a_i are the AR model parameter.

Many different AAR estimation algorithms available are LMS, RLS and Kalman filtering, recursive AR algorithm.

Hjorth parameters:

It indicates the statistical property of a signal in time domain. It has three parameters-

A. Activity: It is the variance of the time function. It indicates the surface of power spectrum in frequency domain.

$$Activity = var(y\ (t\ ))$$

B. Mobility: It is defined as the square root of the ratio of the variance of the first derivative of the signal and that of the signal.

$$Mobility = \sqrt{\frac{var\,(y'\,(t))}{var\,(y\,(t))}}$$

C: Complexity: It indicates how the shape of a signal is similar to a pure sine wave. The value of Complexity converges to 1 as the shape of signal gets more similar to a pure sine wave.

$$Complexity = \frac{mobility\ \ (y'\,(t))}{mobility\ \ (y\,(t))}$$

Power spectral density:

EEG signal is decomposed into various frequency bands (Alpha, Beta, Theta, Delta) by applying Fast Fourier Transform (FFT). Applying FFT yields a complex number for each frequency bin, from which we can extract the amplitude and the phase part of the signal for specific frequency. Power spectral density is computed as the square of the magnitude of FFT signal and is expressed in micro-volts^2 per Hertz (μ-V^2/Hz).

Band power estimate:

Average band power yields the contribution of a given frequency band to overall power of the signal. The average bandpower is also a very relevant metric for sleep research because it allows to differentiate between the different sleep stages. For instance, deep sleep is characterized by its predominance of slow-waves with a frequency range comprised between 0.5 to 4 Hz (i.e. delta band), which reflects a synchronized brain activity. Conversely, wakefulness is characterized by very little delta activity and much more higher-frequencies activity. Therefore, to compute the delta bandpower for both deep sleep and wakefulness, the former would be very high and the latter very low.

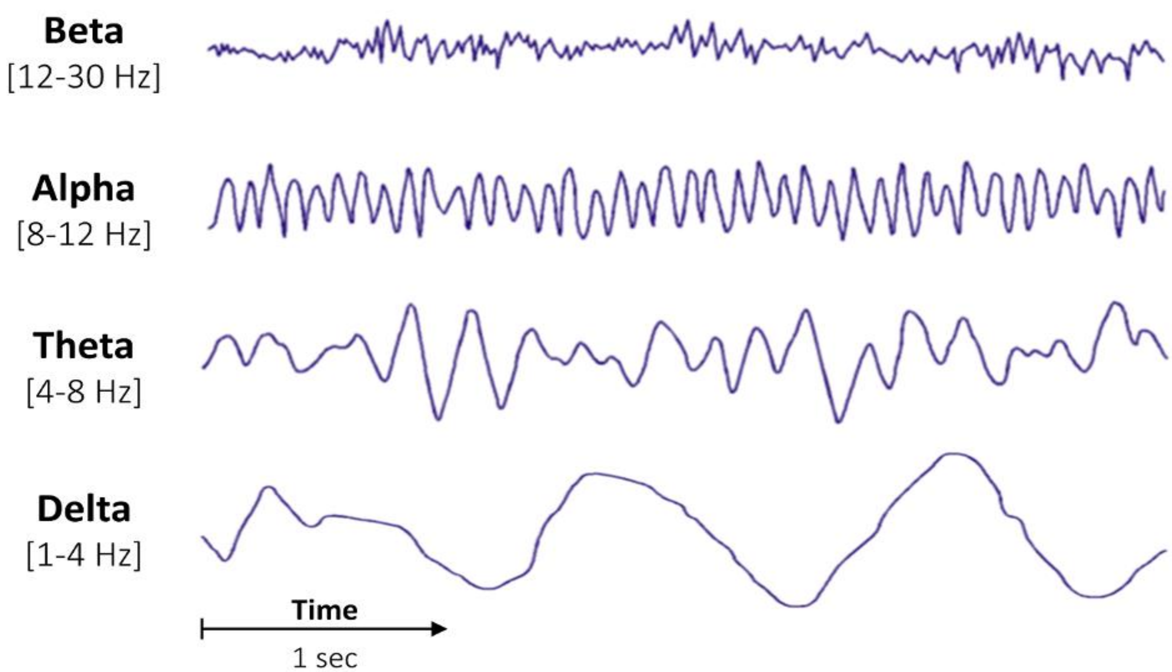


Fig 1.18: Different frequency bands of EEG signal

<u>Wavelet coefficient(WT):</u>

It is used for signal preprocessing or denoising and for feature extraction. It is a powerful signal processing technique which overcomes the shortcomings of other methods such as the Fourier transform. It provides a smooth representation in comparison to windowed representation in the short time Fourier transform (STFT). So even very minute details, sudden changes and similarities in the EEG signals can be detected. It has the capability to analyze EEG signals at different scales. Examples of mother wavelets used for EEG processing:

(a) Daubechies (db)

(b) Morlet

(c) Biorthogonal (bior)

(d) Orthogonal Cubic Spline (ocs)

(e) Mexican Hat (MH)

(f) Haar

(g) Complex Gaussian (CG)

(h) Coiflet (coif) wavelet.

The wavelet decomposition of a noisy signal concentrates intrinsic signal information in a few wavelet coefficients having large absolute values without modifying the random distribution of noise. Hence denoising can be achieved by thresholding the wavelet coefficients.

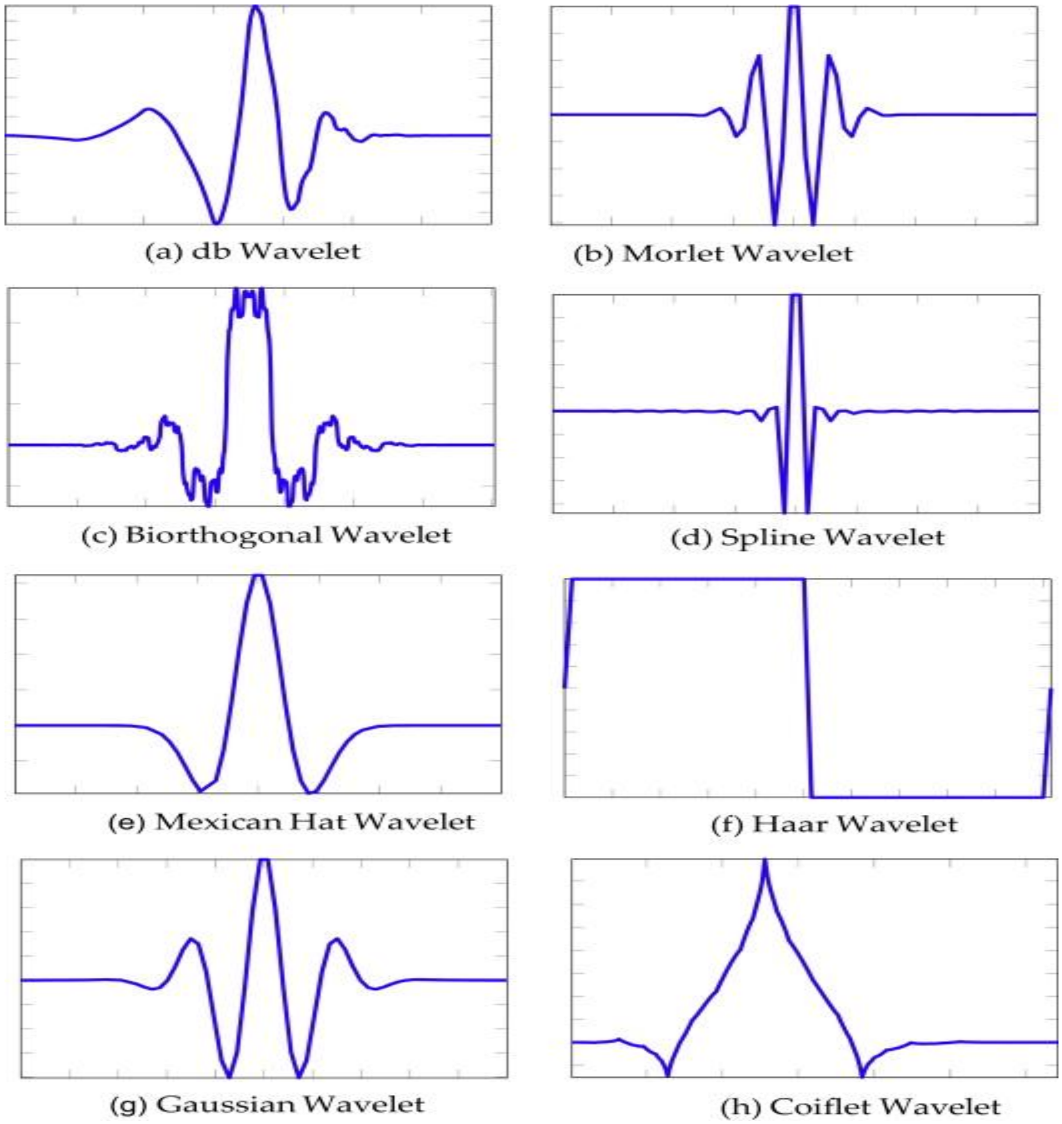


Fig 1.19: Examples of wavelets used for EEG processing

Approximate entropy:

The approximate entropy (ApEn) concept was developed by Pincus. It gives a measure of system complexity. A high value of ApEn indicates random and unpredictable variation, whereas a low value of ApEn indicates regularity and predictability in a time series. ApEn has the capability to differentiate among a wide variety of systems, including nonlinear deterministic, stochastic and noisy systems, while being applicable to medium-sized time series. The application of ApEn includes the analysis of physiological signals. For instance, it was used to recognize epileptic activity.

In mathematical terms, ApEn is derived from the correlation integral $C_{m,i}(r)$. As described by Pincus, the ApEn is computed as:

$$ApEn\,(N, m, r) = \Phi^{m}\,(r) - \Phi^{m+1}(r)$$

$$\Phi^{m}(r) = (N-(m-1))^{-1} \sum_{i=1}^{N-(m-1)} ln\, c_{m,i}\,(r)$$

$$ApEn(N, m, r = (N-(m-1))^{-1} \sum_{i=1}^{N-(m-1)} lnC_{m,i}(r) - (N-m)^{-1} \sum_{i=1}^{N-m} lnC_{m+1,i}(r)$$

ApEn has the following parameters: the vector length m, the filter factor r, and the number of data points N. The value of N for the ApEn computation is typically between 75 and 5000. ApEn measures the logarithmic likelihood that sets of patterns that are close for m-observations remain close on the next incremental comparisons. ApEn signifies how different segments of the signal with similar recent histories remain similar in the future. As ApEn decreases, the complexity of the signal decreases and determinism increases.

Common spatial patterns:

This method was suggested for classification of multi-channel EEG during imagined hand movements by H. Ramoser. The main idea is to use a linear transform to project the multi-channel EEG data into low dimensional spatial subspace with a projection matrix, of which each row consists of weights for channels. This transformation can minimize the variation of two class signal matrices. This method is based on simultaneous diagonalization of covariance matrices of both classes.

Let $\mathbf{X_H}$ and $\mathbf{X_F}$ denote the preprocessed EEG matrices under two conditions(hand and foot) with dimensions NXT, where N is the no of channels and T is the no of samples per channel. The normalized spatial covariance of EEG can be represented as

$$R_H = \frac{X_H X_H^T}{trace(X_H X_H^T)} \qquad\qquad R_F = \frac{X_F X_F^T}{trace(X_F X_F^T)}$$

where, $\mathbf{X^T}$ is the transpose of **X** and trace(**A**) is the sum of the diagonal elements of **A**.

1.7: Importance of Feature Selection in EEG-based BCI:

Feature Selection is one of the core concepts of Machine Learning which impacts the performance of the model significantly. Having irrelevant or redundant features in the dataset affects the performance of model adversely i.e., it not only decreases the accuracy of the model but also makes the model learn on insignificant features, and also increases the training time. Therefore, data cleaning by selecting appropriate features should be the foremost criterion of model designing.

Feature selection (also known as subset selection) is defined as the process where we select only those features that contribute the most to the predicted output. Performing feature selection provides the following benefits-

A. Reduces Overfitting: unnecessary resource allocation for irrelevant dataset is avoided
B. Improves Accuracy: Reduces the complexity of the model and performance improves
C. Less training time: Fewer data points reduce the high-dimensionality of the model and hence requires less time to get trained

The different types of general feature selection methods –

A. Filter methods
B. Wrapper methods
C. Embedded methods

Often feature selection is confused with dimensionality reduction, since both the methods tend to reduce number of features in a dataset. However both are distinct. A dimensionality reduction method does so by creating new combinations of attributes in contrast feature selection methods include and exclude attributes present in the data without changing them.

1.8: Role of Evolutionary Optimization in case of Feature Selection of EEG-based BCI:

Real-world problems of Machine Learning and Data Mining comprises of huge number of datasets. The datasets basically represents the features of the problem. However, many of these features are redundant or irrelevant and using them contributes no value. In fact, using these features lowers the performance of the algorithm. Hence is the need of feature selection. By eliminating the redundant data, the huge dimensionality curse can be avoided. Also feature selection helps in increasing the speed of the learning process, simplifies the learning model and improves the performance.

In reality, feature selection is a difficult task because of the enormous search space it involves. If there are n feature in a dataset, the total number of possible search space is 2^n. In case of complex problems, with large value of n the task of feature selection becomes more difficult to achieve. Large search space often leads to feature interaction which in turn leads relevant data to become irrelevant and vice-versa. To solve this problem a lot of search techniques such as greedy search, complete search, heuristic search, random search are thought of. But these search techniques involves high computational cost and suffer from stagnation in local optima. Hence comes the importance of evolutionary computation techniques which are well-known for their global search ability.

Some of the evolutionary computation techniques involve Genetic Algorithms, Genetic Programming, Differential Evolutionary Optimization, Particle Swarm optimization, Ant colony Optimization etc. All these evolutionary techniques constitute a different approach but all are based on the principle of natural evolution. Evolutionary optimization techniques generally aim at finding a given number of pareto-optimal solutions. These solutions are distributed uniformly in pareto –optimal front, which gives the decision maker insight into the problem so that a particular conclusion can be reached.

The advantage of using an evolutionary optimization algorithm is that it reduces computational complexity and increases feature set effectiveness by selecting relevant features. Besides these evolutionary techniques do not need any domain knowledge. Also they do not make assumption of the search spaces. These techniques being population-based produces multiple solutions in one run and hence are most suitable for multi-objective feature selection.

1.9: Overview of the Thesis:

This thesis is an attempt to propose a method of feature selection of BCI dataset using the Differential evolutionary optimization algorithm and to obtain the accuracy of classification by using supervised learning classifier.

The thesis is organized into four chapters. While chapter 1 discusses the basic concepts and gives us a detailed idea of Brain Computer interfacing, chapter 2 is about Differential Evolutionary optimization. It gives the general idea about the algorithm, the steps and the pseudo-code of the algorithm. It also gives an idea of the research works done with the algorithm.

Chapter 3 provides detailed insights of the work done. It explains the detailed methodology adopted to solve the problem. It also includes the Matlab code of the algorithm and the results obtained.

Chapter 4 is the concluding chapter of the thesis. It introduces the reader to the future avenues in the world of optimization that can be achieved with Differential Evolutionary algorithm.

1.10: References:

[1] Eric C. Leuthardt, Gerwin Schalk, Jarod Roland, Adam Rouse, Daniel W. Moran, "Evolution of brain-computer interfaces: going beyond classic motor physiology." July 2009; Neurosurg Focus., 27(1): E4.

[2] Akshaya R. Mane , Prof. S. D. BiradarProf. R. K. Shastri, "Review paper on Feature Extraction Methods for EEG Signal Analysis." 2015

[3] Saugat Bhattacharyya, Poulami Ghosh, Anwesha Khasnobish, Ankita Mazumder, D. N. Tibarewala, "A Review on Brain Imaging Techniques for BCI Applications ." January 2017

[4] Bing Xue, Mengjie Zhang, Will N. Browne, Xin Yao, "A Survey on Evolutionary Computation Approaches to Feature Selection." DOI 10.1109/TEVC.2015.2504420, IEEE

[5] Tzyy-Ping Jung, "Principles and Applications of Brain-Computer Interfaces."

[6] Kai Keng Ang, Cuntai Guan, "Brain–Computer Interface for Neurorehabilitation of Upper Limb After Stroke." Proceedings of the IEEE ,Volume: 103 , Issue: 6 , June 2015

[7] Sarah N. Abdulkader , Ayman Atia, Mostafa-Sami M. Mostafa, "Brain computer interfacing: Applications and challenges." Egyptian Informatics Journal Volume 16, Issue 2, p 213-230, July 2015

Chapter 2: Differential Evolution

2.1: Introduction:

Differential evolution (abbreviated as DE) was first proposed in 1995 by Storn and Price at Berkeley. DE can be seen as a typical evolutionary algorithm because it closely follows the "survival of the fittest" principle.

DE is a stochastic, direct search optimization method which optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality (cost function or fitness function). DE optimizes a problem by maintaining a population of candidate solutions and creating new candidate solutions by combining existing ones according to its simple formulae, and then keeping whichever candidate solution has the best score or fitness on the optimization problem at hand. If the new position of an agent is an improvement then it is accepted and forms part of the population, otherwise the new position is simply discarded. The process is repeated and by doing so it is hoped.

DE is used for multidimensional real-valued functions but does not use the gradient of the problem being optimized, which means DE does not require the optimization problem to be differentiable. DE can therefore also be used on optimization problems that are not even continuous, are noisy, change over time, etc.

This algorithms fall under meta-heuristics since they make few or no assumptions about the problem being optimized and can search very large spaces of possible solution elements. DE do not guarantee an optimal solution is ever found.

It is generally considered as accurate, reasonably fast and robust optimization method. The main advantages of DE are finding the true global minimum regardless of the initial parameter values, fast convergence, and using a few control parameters. The main characteristic of DE is with its ability to search with floating point representation instead of binary representation as used in many basic EAs such as GAs. The characteristics together with other factors of DE makes it a fast and robust algorithm as an alternative to EA and have found an increasing applications in a number of engineering areas including power engineering.

2.2: Literature review of Differential Evolution:

Researchers from various fields across the globe have been carrying out to a lot of research work to improve the performance of DE either by tuning its control parameters or by inventing different variants of DE. The DE variants that were developed over the past decade have appeared to be competitive against the existing best-known real parameter optimizers.

The various DE variants differ on the basis of the target vector selected, the number of difference vectors used, and the way that crossover points are determined. A general notation is DE/x/y/z where x is the method of target vector selection, y is the number of difference vector used and z is the crossover method used.

Fan and Lampinen proposed a trigonometric mutation with a probability of τ and the mutation scheme of DE/rand/1 with a probability of $(1-\tau)$, which improved the performance of DE.

A DE/rand/1/either-or algorithm was proposed by Price *et al* where the pure mutants occur with a probability p_F and those that are pure recombinants occur with a probability$1 - p_F$. It showed better performance compared to classical DE-variants rand/1/bin and target-to-best/1/bin. This method provided an efficient way way to implement the dual axis search in the *k*-*F* plane (*k* indicating the combination coefficient of the arithmetic crossover and *F* being the scale Factor). Rahnamayan *et al.* proposed an Opposition Based Differential Evolution (ODE) for faster global search and optimization, which finds wide applications in noisy optimization problems. The concept was to use opposition number based optimization in three levels, namely, population initialization, generation jumping, and local improvement of the population's best member.

DEGL (DE with Neighborhood-Based Mutation) was proposed by Das *et al.* that balance between explorative(it introduces new information into the population) and exploitative(ability of a search algorithm to use the information already collected thus orienting the search toward the goal) tendencies. In DEGL, vectors neighbourhood (a set of parameter vectors) is considered when updating its position.

Qin *et al*. proposed a self-adaptive variant of DE (SaDE) where the control parameter values and the trial vector generation strategies are self adapted by previous experiences in generating promising solutions. In SaDE, for each target vector in the current generation, a trial vector generation strategy is selected from the candidate pool based on the probability learned from its success rate in generating improved solutions within a certain number of previous generations which is known as the learning period (LP). The selected strategy is then applied to the target vector for generating the trial vector. For each generation, the probabilities of choosing a strategy in the candidate pool are summed to 1. Initially, the probabilities are equal and are subsequently adapted over LP generations according to success and failure rates.

Four effective trial vector generation strategies in SaDE are DE/rand/1/bin, DE/rand-to-best/2/bin, DE/rand/2/bin and DE/current-to-rand/1. The first three variants are associated with binomial type crossover while the last one uses arithmetic recombination.

JADE, an adaptive DE-variant, implements a new mutation strategy DE/current-to-pbest and uses an optional external archive to track the previous history of failure and success. The control parameters are updated in an adaptive manner with generations.

Researchers have also hybridized DE (combined the best features of two or more algorithms forming a new algorithm to outperform its ancestors over application-specific or general benchmark problems) with other global optimization algorithms like PSO, ant colony systems, artificial immune systems (AIS), bacterial foraging optimization algorithm (BFOA) and simulated annealing (SA).

Hendtlass proposed a hybrid optimizer based on PSO and DE, where particle positions gets updated if their offspring possess better fitness.

DEPSO, a popular hybrid algorithm of DE and PSO was proposed by Zang and Xie. In DEPSO, the PSO algorithm and the DE operator alternate at the odd iterations and at the even iterations.

PSO-DV (particle swarm optimization with differentially perturbed velocity), a synergy of PSO and DE was proposed by Das *et al*. It introduces a differential operator borrowed from DE in the velocity-update scheme of PSO and is based on the concept that a particle shifts to a new location if the new location has better fitness value.

Moore et al. proposed hybridization of DE/rand/2/bin and modified PSO with "Ring" topology.

Chemotactic differential evolution (CDE), a hybrid algorithm of DE with BFOA (an optimization process where an animal seeks to maximize energy per unit time spent for foraging) was proposed bt Biswas *et al*.

The attraction-repulsion concept of electromagnetism-like algorithm to boost the mutation operation of the original DE was proposed by Ali *et al*.

Noman et al. proposed a synergy of DE with Local Search Methods . The length of the local search is adjusted adaptively using a hill-climbing heuristic.

NSDE, a hybridization of DE with neighborhood search was proposed by Yang *et al.*

A Self-adaptive NSDE (SaNSDE), capable of optimizing large-scale non-separable problems was also proposed by Yang *et al*. It proposes three self -adaptive strategies- self-adaptive choice of the mutation strategy between two alternatives, self-adaptation of the scale factor F, and self-adaptation of the crossover rate Cr.

A discrete version of DE, known as DDE, for the no-wait flowshop scheduling problem with total flow time criterion was proposed by Tasgetiren *et al*.

A DE variant which can operate in binary problem spaces without diverging from the basic search mechanism of the classical DE, was proposed by Pampara *et al*.

Several researchers also parallelized DE to improve its speed and accuracy on expensive optimization problems. Lampinen's parallelization scheme employed the entire population to be kept in a master processor and selects individuals for mating and sends them to slave processors for performing other operations.

Another parallel DE scheme was proposed by Tasoulis *et al*. where a whole subpopulation was mapped to a processor, allowing different subpopulations to evolve independently toward a solution.

Besides these DE variants, a lot of research work has been done on DE by tuning its control parameters mainly to improve its speed and accuracy. The three control factors of DE are the mutation scale factor F, the crossover constant Cr, and the population size NP.

"jDE", a self-adaptation scheme for the DE control parameters, was proposed by Brest *et al*. Each individual in the population was assigned a set of F and Cr values. The better values of the control parameters lead to better individuals that in turn which are likely to survive and produce offspring.

An adaptation strategy for DE known as ADE was proposed by Zaharie *et al*. It is based on the concept of controlling the population diversity and implemented a multi-population approach.

2.3: Main Steps of Differential Evolutionary Algorithm:

The main steps of the DE algorithm are given below:

A. Initialization
B. Mutation
C. Crossover
D. Selection

Initialization: All the parameters required to run the code are introduced. The parameters include upper bound(X_{max}), lower bound (X_{min}), population size (NP), cross-over ratio (Cr), scaling factor (F), number of generations (G), number of features to be selected (D). It begins with a randomly initiated population of *NP D* dimensional real-valued parameter vectors. Each vector (known as genome or chromosome) forms a candidate solution to the optimization problem. The subsequent generations in DE are denoted by $G = 0, 1 \ldots, G_{max}$. Since the parameter vectors are likely to be changed over different generations, the i^{th} vector of the population at the current generation is denoted as:

$$\vec{X}_{i,G} = [x_{1,i,G}, x_{2,i,G} x_{3,i,G} \ldots \ldots \ldots x_{D,i,G}]$$

For each parameter of the problem, there is a certain range within which the value of the parameter is restricted, often because parameters are related to physical components or measures that have natural bounds (for example if one parameter is a length or mass, it cannot be negative). The initial population (at $G = 0$) should cover this range as much as possible by uniformly randomizing individuals within the search space constrained by the prescribed minimum and maximum bounds: $\vec{X}_{min} = \{x_{1,min}, x_{2,min}, \ldots, x_{D,min}\}$ and $\vec{X}_{max} = \{x_{1,max}, x_{2,max}, \ldots, x_{D,max}\}$. Therefore, the j^{th} component of the i^{th} vector is initialized as:

$$x_{j,i,0} = x_{j,min} + rand_{i,j}[0, 1] \cdot (x_{j,max} - x_{j,min})$$

where $rand_{i,j}[0, 1]$ is a uniformly distributed random number lying between 0 and 1 (actually $0 \le randi,j[0, 1] \le 1$) and is instantiated independently for each component of the *i*-th vector.

<u>Mutation:</u> The term "mutation" is mostly associated with biology and it refers to the sudden change in gene characteristics of a chromosome. In context of DE it refers to perturbation with a random element. In DE-literature, a parent vector from the current generation is called target vector, a mutant vector obtained through the differential mutation operation is known as donor vector and finally an offspring formed by recombining the donor with the target vector is called trial vector. The donor vector for each i^{th} target vector from the current population, is created by three other distinct randomly chosen parameter vectors. Considering, $\vec{X}_{r_1^i}, \vec{X}_{r_2^i}, \vec{X}_{r_3^i}$ are randomly chosen from the current population where the indices r_1^i, r_2^i, r_3^i are mutually exclusive integers randomly chosen from the range
[1, *NP*], which are also different from the base vector index *i*. These indices are randomly generated once for each mutant vector. Now the difference of any two of these three vectors is scaled by a scalar number *F* (that typically lies in the interval [0.4, 1]) and the scaled difference is added to the third one to obtain the donor vector $\vec{V}_{i,G}$. The donor vector is as follows:

$$\vec{V}_{i,G} = \vec{X}_{r_1^i,G} + F.(\vec{X}_{r_2^i,G} - \vec{X}_{r_3^i,G})$$

<u>Crossover:</u> The role of the crossover operation is to increase population diversity after the generation of donor vector through the mutation process. The donor vector exchanges its components with the target vector $\vec{X}_{i,G}$ under this operation to form the trial vector $\vec{U}_{i,G} = [u_{1,i,G}, u_{2,i,G}, u_{3,i,G}, \dots \dots \dots u_{D,i,G}]$. The DE family of algorithms can use two kinds of crossover methods—exponential (or two-point

modulo) and binomial (or uniform). In this particular study, only binomial crossover is considered. It is performed on each of the *D* variables whenever a randomly generated number between 0 and 1 is less than or equal to the *Cr* value. In this case, the number of parameters inherited from the donor has a (nearly) binomial distribution. The binomial crossover operation is as follows:

$$u_{j,i,G} = \begin{cases} v_{j,i,G}\ , & if\ rand_{i,j}[0,1] \leq Cr\ or\ j = j_{rand} \\ & x_{j,i,G}\ , otherwise \end{cases}$$

where, $rand_{i,j}[0, 1]$ is a uniformly distributed random number, which is called anew for each j^{th} component of the i^{th} parameter vector. $j_{rand} \in [1, 2, ..., D]$ is a randomly chosen index, which ensures that $\vec{U}_{i,G}$ gets at least one component from $\vec{V}_{i,G}$.

<u>Selection:</u> The selection step of DE is to determine whether the target or the trial vector survives to the next generation, i.e., at $G = G + 1$. The selection operation is as follows:

$$\begin{aligned} \vec{X}_{i,G+1} &= \vec{U}_{i,G}\ if\ f(\vec{U}_{i,G}) \leq f(\vec{X}_{i,G}) \\ &= \vec{X}_{i,G}\ if\ f\left(\vec{U}_{i,G}\right) > f(\vec{X}_{i,G}) \end{aligned}$$

where, $f(\vec{X})$ is the objective function to be minimized. if the new trial vector yields an equal or lower value of the objective function, the corresponding target vector is replaced in the next generation; otherwise the target is kept in the population. Hence, the population either gets better (with respect to the minimization of the objective function) or remains the same in fitness status, but it never deteriorates.

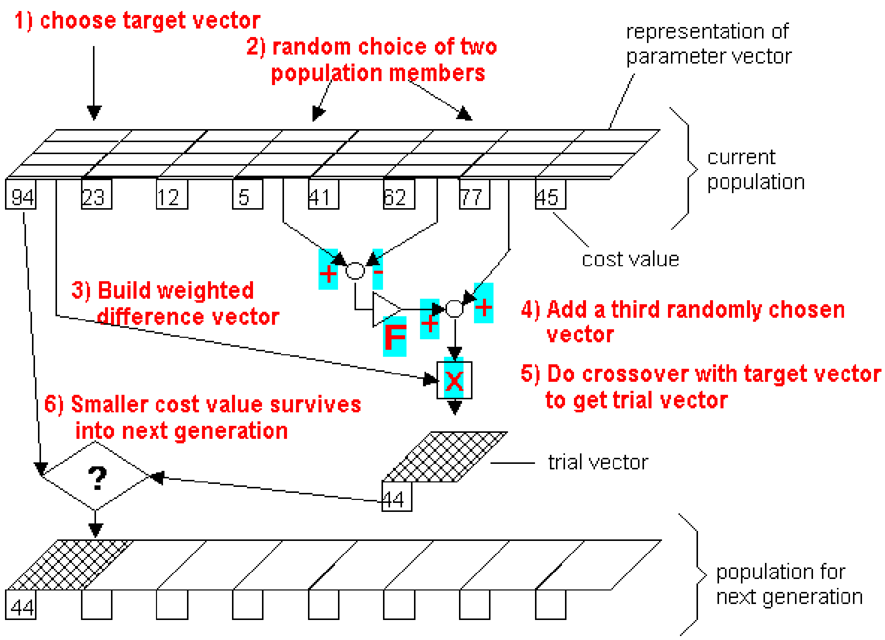


Fig 2.1: Working principle of Differential Evolutionary Algorithm

2.4: Pseudo-code of Differential Evolutionary Algorithm:

Step 1: The values of the control parameters of DE: scale factor *F*, crossover rate *Cr*, and the population size *NP* are initialized from user.

Step 2: The generation number *G* = 0 is set and randomly initialization of a population of *NP* individuals $P_G = \{\vec{X}_{1,G}, \dots \dots \dots \dots \dots, \vec{X}_{NP,G}\}$ with $\vec{X}_{i,G} = \{x_{1,i,G}, x_{2,i,G}, x_{3,i,G} \dots \dots \dots x_{D,i,G}\}$ and each individual uniformly distributed in the range $[\vec{X}_{min}, \vec{X}_{max}]$, where $\vec{X}_{min} = \{x_{1,min}, x_{2,min}, \dots \dots .. x_{D,min}\}$ and $\vec{X}_{max} = \{x_{1,max}, x_{2,max}, \dots \dots \dots x_{3,max}\}$ with *i* = [1, 2, ….,*NP*] is done.

Step 3. WHILE the stopping criterion is not satisfied

DO

FOR *i* = 1 to *NP* //do for each individual sequentially

Step 2.1 Mutation Step

Generate a donor vector $\vec{V}_{i,G} = \{v_{1,i,G}, \dots \dots \dots., \}$ $\{v_{D,i,G}\}$ corresponding to the i^{th} target vector $\vec{X}_{i,G}$ via the differential mutation scheme of DE as:

$$\vec{V}_{i,G} = \vec{X}_{r_1^i,G} + F.(\vec{X}_{r_2^i,G} - \vec{X}_{r_3^i,G})$$

Step 2.2 Crossover Step

A trial vector $\vec{U}_{i,G} = \{u_{1,i,G}, \dots \dots \dots u_{D,i,G}\}$ for the i^{th} target vector $\vec{X}_{i,G}$ through binomial crossover is generated in the following way:

$u_{j,i,G} = v_{j,i,G}$, if ($rand_{i,j}[0, 1] \leq Cr$ or $j = jrand$)

$= x_{j,i,G}$, otherwise

Step 2.3 Selection Step

The trial vector $\vec{U}_{i,G}$ is evaluated

IF $f(\vec{U}_{i,G}) \leq f(\vec{X}_{i,G})$, THEN $\vec{X}_{i,G+1} = \vec{U}_{i,G}$

ELSE $\vec{X}_{i,G+1} = \vec{X}_{i,G}$

END IF

END FOR

Step 2.4 The Generation Count is increased

$G = G + 1$

END WHILE

2.5: Applications of Differential Evolutionary Algorithm:

A. Finding predictive Gene subsets in microarray data:

DNA microarray technologies facilitate the monitoring of thousands of gene expressions simultaneously. Classification of samples using microarray data is necessary to decide which genes should be included in the classifier. Including too many genes adds noise and on the other hand selecting too few genes may not give accurate result for the test data. So, DE along with Fuzzy Neural Network (FNN) is used in microarray classification to determine the optimal, or near optimal, or a subset of predictive genes on complex and large spaces of possible gene sets. The algorithm maintains a population of trial gene subsets, imposes random changes on the genes that compose those subsets (mutation), and incorporates selection driven by a neural network classifier to determine the relevant or informative ones. Only those relevant ones are included in successive generations while others are eliminated. Every subset is given as input to an FNN classifier (at each iteration). The effectiveness of the FNN determines the fitness of the subset of genes i.e., FNNs is used as a classifier to evaluate the fitness of each gene subset.

B. Optimization of Strategies based on Financial Time Series:

In financial markets, trading rules determine the future price trend by analysis of previous price movement. Successful trading strategy leads to profit making and provides a good market forecast. With the growing competitiveness in the current market scenario and with the increase in the complexities of the problems, researchers are now relying on various optimization algorithms to ascertain the technical rules. A new version of DE applied to general scheme of a trading strategy yielded better profitability of the trading strategies.

C. Optimal Operation of Multipurpose Reservoir:

The objective of the optimization is to maximize the hydropower production. For this particular problem, the constraints are reservoir capacity, turbine release capacity constraints, irrigation supply demand constraints and storage continuity. Many variants of DE were used among which De/best/1/bin proved to be the best strategy giving the optimal solution. The results of monthly maximized hydropower production and irrigation releases enable the decision maker to take decisions regarding operation policy of the reservoir.

D. Optimization in Transmission Expansion Planning (TEP):

TEP has become more complicated in recent times due to the rapid growth of the transmission networks. The objective is to locate the additional transmission lines that must be added to meet the forecasted load in the system adequately with minimum cost. Dong et al. proposed a differential evolution based method for power system planning problem which aims at finding the minimum possible cost of transmission lines that must be added to meet the forecasted load in a power system. The planning involves several objectives like expansion investment cost, the reliability objective-expected energy (minimizing the investments and operational costs, ensure an adequate quality level of energy supply to customers), the social welfare objective-expected economic losses (economic losses from the unreliable power supply, customer economic losses caused by load shedding) and the system expansion flexibility.

2.6: Conclusion:

The DE family of algorithms which emerged nearly two decades ago, was initially meant meant for global numerical optimization over the continuous search spaces. Later, the field was greatly enhanced by researches from various domains.
It is basically a nature inspired evolutionary technique where the fittest individuals of a population are the ones that produce more offspring, which in turn inherit the good traits of the parents. This makes the new generation more likely to survive in the future, and hence the population improves over subsequent generation. By the mechanism of mutation, recombination and selection operation, DE fids the set of optimal or near optimal solution to a specific function.

Over time, various variants of DE like SaDE, parallel DE, hybrid DE, discrete and binary forms of DE etc. were proposed by various researchers to improve the convergence speed of conventional DE and to meet various constraint criterions.

The DE algorithms can use both synchronous and asynchronous modes of survivor selection or population update. Unlike other evolutionary computation techniques, basic DE proves to be a very simple algorithm whose implementation requires only a few lines of code in any standard programming language. Also, the control parameters in DE are very less few precisely 3: the scale factor, the crossover rate and the population size. All these factors combined together makes DE the best choice for single objective, constrained, dynamic, large-scale, multi-objective, and multi-modal optimization problems.
Nowadays countless new algorithms are being published almost every day deriving inspirations from nature and ranging from various sources like human beings to flu virus. But unlike these algorithms, which provide a single and somewhat narrowed

search strategy due to their source of inspiration, DE provides the user with a flexible set of offspring generation strategies. This strategy provides the user with strong intuitive justifications behind the resulting search moves. Due to this flexibility, DE serves as a versatile and robust optimizer in widely differing and difficult optimization scenarios. It is believed that DE would continue to remain a vibrant and active field of multi-disciplinary research in subsequent years.

2.7: Reference:

[1] Swagatam Das, Ponnuthurai Nagaratnam Suganthan, "Differential Evolution: A Survey of the State-of-the-Art." IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 15, NO. 1, FEBRUARY 2011

[2] Dervis Karaboga, Selcuk Okdem, " A Simple and Global Optimization Algorithm for Engineering Problems: Differential Evolution Algorithm." J Elec Engin, VOL.12, NO.1 Turkey,2004

[3] O. G. Monakhov, E.A.Monakhova, and M. Pant, "Application of Differential Evolution Algorithm for Optimization of Strategies Based on Financial Time Series." ISSN 1995-4239, Numerical Analysis and Applications, 2016, Vol. 9, No. 2, pp. 150–158. c Pleiades Publishing, Ltd., 2016.

[4] Dattatray Gangaram Regulwar, Sumant Anant Choudhari, Pallakury Anand Raj, "Differential Evolution Algorithm with Application to Optimal Operation of Multipurpose Reservoir." J. Water Resource and Protection, 2010, 2, 560-568, June 2010

[5] Ibrahim Al hamrouni, Azhar B khairuddin , M.Salem, "Application of Differential Evolution Algorithm in Transmission Expansion Planning." Applied Mechanics and Materials, Volume: 394, September 2013

[6] Swagatam Das , Sankha Subhra Mullick , and P. N. Suganthan," Recent Advances in Differential Evolution – An Updated Survey." Swarm and Evolutionary Computation,Volume 27, p 1-30, April 2016

[7] D.K. Tasoulis, V.P. Plagianakos, M.N. Vrahatis, "Differential Evolution Algorithms For Finding Predictive Gene Subsets in Microarray Data." IFIP International Conference on Artificial Intelligence Applications and Innovations pp 484-491, 2006

Chapter 3: Differential Evolution induced Feature Selection

3.1: Introduction:

This study is aimed at reducing the dimension of the feature vector by using one of the Evolutionary optimization techniques known as Differential Evolution and a supervised learning classifier.

Before going into the details of the study, a brief knowledge of machine learning is given.

A brief on Machine Learning .....

Machine learning (ML) is a sub-field of Artificial Intelligence (AI) that aims at making systems that can automatically learn and improve without being explicitly programmed. Some of the ML methods are:

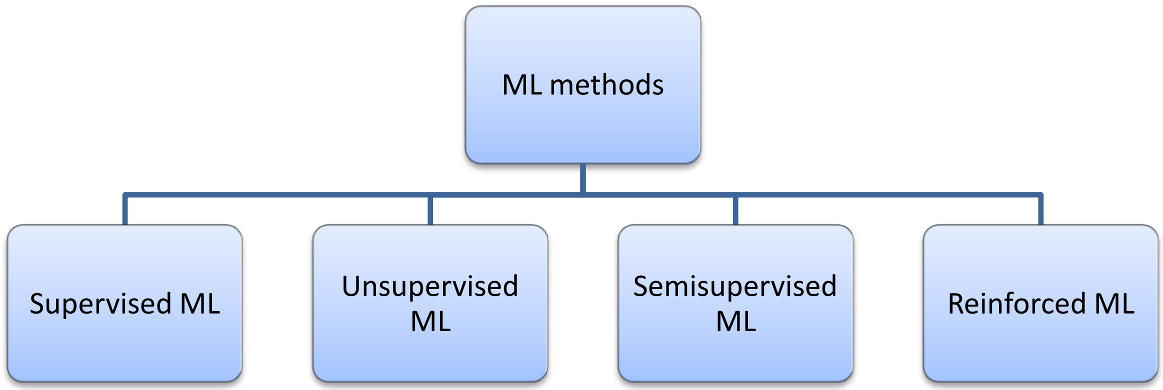


Chart 3.1: Different kinds of ML methods

Supervised ML: The supervised learning algorithms learn from a known dataset (which is also referred as training dataset) and predict the output associated with new inputs (referred as testing dataset).

Unsupervised ML: The unsupervised learning algorithms learn from the data that is neither classified nor labeled. It takes a dataset that contains only inputs and finds structure in the data, like grouping or clustering of data points.

Semi-supervised ML: It falls somewhere between Supervised ML and Unsupervised ML. It uses both labeled (typically a small amount) and unlabeled data (a large amount) for training.

Reinforced ML: The reinforced ML algorithms interact with its environment by producing actions and discover errors or rewards. Reinforced MLs are characterized by trial and error search and delayed reward.

A brief on Support Vector machine ……

In this study, support vector machine is employed to perform linear classification. A Support Vector Machine(SVM) is a supervised learning model, used maily for classification and regression tasks. Considering a given a set of training examples, each marked as belonging to one or the other of two categories, an SVM constructs a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.

A training dataset of n points is given in the form:

$(\vec{x}_1, y_1)$,……………..,$(\vec{x}_n, y_n)$

Where y_i takes the value of either 1 or -1 and indicates the class to which $\vec{x}_i$ belongs. The task is to find the maximum-margin hyperplane which divides the group of points $\vec{x}_i$ for which $y_i = 1$ from the group of points for which $y_i = -1$ is defined so that the distance between the hyperplane and the nearest point $\vec{x}_i$ from either group is maximized. Any hyperplane can be written as the set of points $\vec{x}$ satisfying:

$$\vec{w}.\vec{x}\text{-b}=0 \qquad (\vec{w} \text{ is the normal vector to the hyperplane})$$

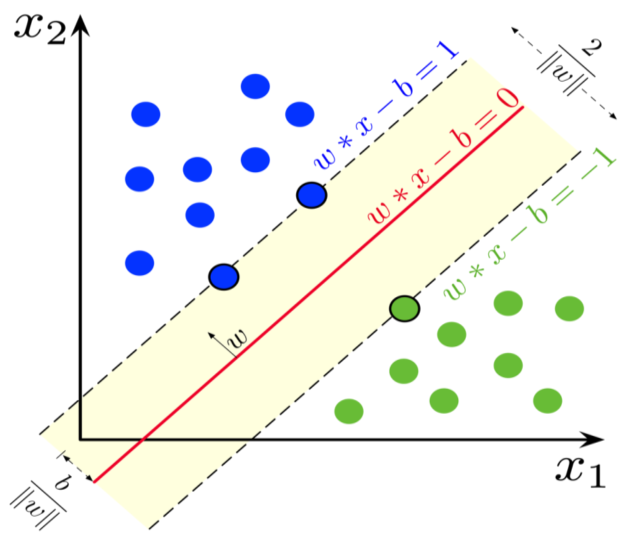


Fig 3.1: Binary SVM (samples on the margin are called the support vectors)

DE Induced Feature Selection………

Considering, there exists 'N' number of data points, the number of ways in which features can be selected from that dataset is $^{N}C_{2}$. So, there must be a way in which we can choose the features in optimum way. Otherwise, working with such huge dataset would increase the computational cost and would contain many irrelevant and redundant features. So, in order to choose the optimum features an objective function or fitness function must be defined.

Using the DE-based learning framework, a number of trial solutions containing pre-defined number of features are generated. Then a pseudo-dataset is constructed from the original dataset where each data point consists of selected features from the solution of the algorithm. Then, we evaluate the precision of each of the selected features with the classification accuracy obtained by testing the learning classifier. After that through the mutation and selection operation of DE, the bad solutions get eliminated and the best solution dominates the population. Finally when the fittest solution represents the near optimal partitioning the dataset with respect to the employed validity index, the evolution of the solution comes to a halt. In this method, the optimal number of features is identified by the evolutionary algorithm.

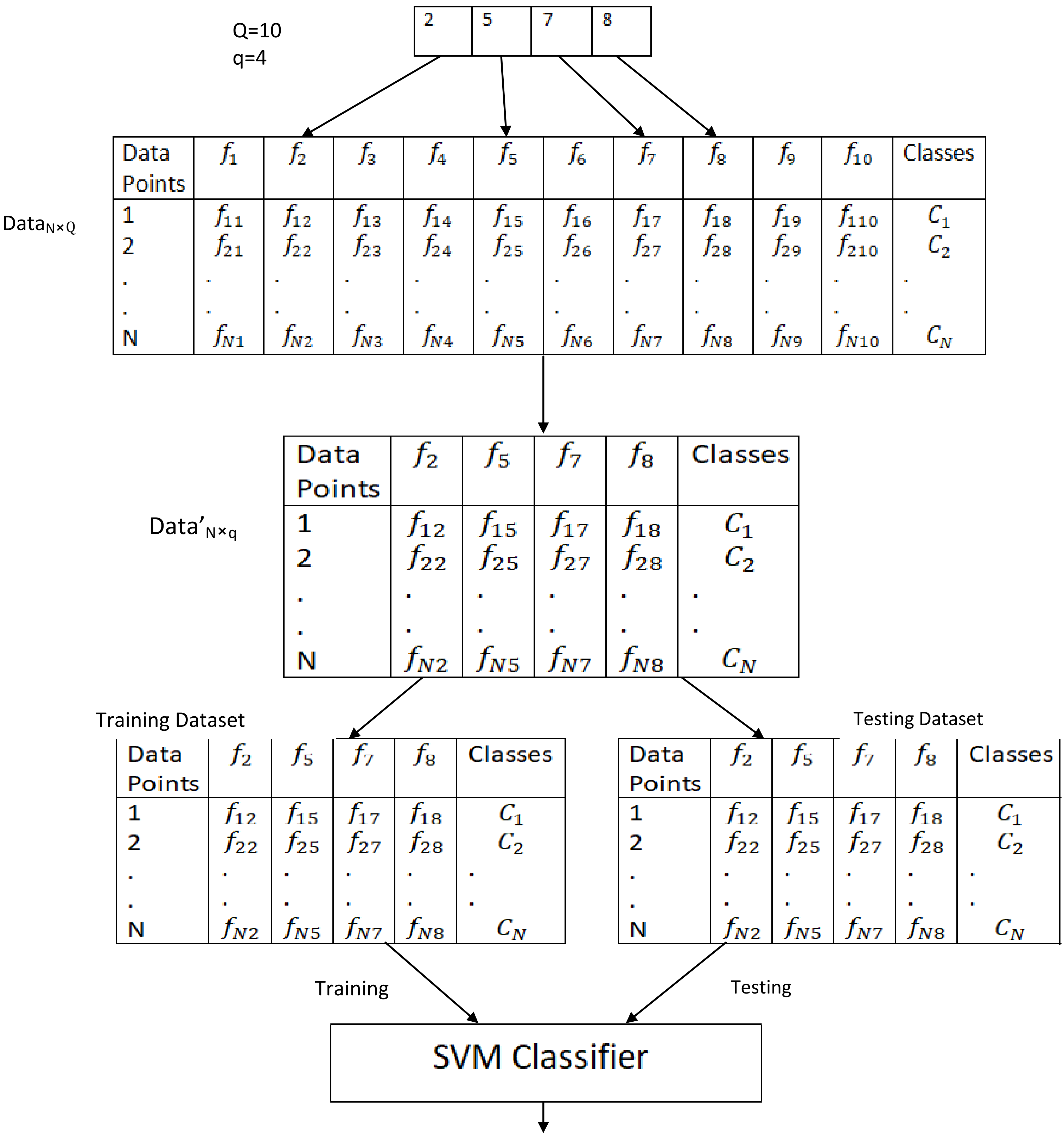
3.2: Proposed Methodology:

Considering $data_{P\times Q}$={$\overrightarrow{data}_1, \overrightarrow{data}_2, \dots\dots\dots\dots, \overrightarrow{data}_P$} be a dataset with P data-points and Q number of features, each data-points comprising of an assigned class label l∈[1,M] for M classes. The DE algorithm finds out an optimal set of q features where q<Q. Since features from the dataset can be selected in any number of ways, an objective function is defined to select q number of features from the original Q number of features. In other words, the problem simplifies to finding a set of q features (from Q features) of optimal or near-optimal adequacy, as compared to all other feasible solutions.

In order to evaluate the quality of the feature selection, a pseudo-dataset $data\ '_{P\times q}$ is created from the original dataset$data_{P\times Q}$. The dataset $data\ '_{P\times q}$ is comprised of all P data-points and only the selected q features.

Now, the dataset $data\ '_{P\times q}$ is decomposed into two parts- training dataset and testing dataset. The training dataset along with their class label are fed into SVM classifier. After training, the testing datasets are also fed to the SVM classifier. Since, the nominal class of the testing dataset is known to the user, the mean number of correctly classified data-points is evaluated. In other words, the classification accuracy of the features is obtained. The classification accuracy obtained by the SVM classifier is used for fitness function evaluation.

As an example, let us consider a solution encoding scheme $data_{N\times Q}$ having Q=10 features. The features are denoted by $f_1, f_2, \dots\dots\dots. f_{10}$ respectively. Let's suppose that the second, fifth, seventh and eighth features have come out as the optimal solution of DE. That is, the f_2, f_5, f_7, f_8 have been activated for q=4. So now, the dataset is reduced to 4 features and N data-points. We call this reduced dataset as $data'_{N\times q}$ where q=4. Next, we decompose this dataset is decomposed into two subsets testing data training dataset and testing dataset. First the testing and then the training datasets are then fed into SVM classifier. Then the accuracy of feature selection is obtained. The classification accuracy serves as the cost function of the DE algorithm.



Cost Function = Classification Accuracy

Fig 3.1: Diagrammatic representation of proposed methodology

3.3: Experiments & Results:

Dataset Description:

The dataset was generated at the Artificial Intelligence Laboratory of Jadavpur University. The dataset is basically an EEG signal obtained from an individual. The recording of the EEG signal has been done using BrainAmp Amplifiers and a 128 Ag/AgCl electrode cap from ECI. 118 channels were used for the measurement of the EEG signals and the signals were obtained by following the positions of the international 10/20 electrode system.

At first, the subject was made to focus on fixation cross for 3 seconds, then the subject concentrated on a dim bulb light for 5 seconds, after that a rest period was given for subsequent 10 seconds and finally again the subject concentrated on the light for next 5 seconds.

The dataset which have been used in this study is the EEG signals when the subject concentrated on the light for different periods of time.

MATLAB CODE:

```
clear all;
close all;
clc;

NP=100;                              %Population size%
D=5;                    %Dimension of solution = No. of optimal
features%
CR=0.96;                             %Crossover ratio%
F=0.5;                               %Scaling Factor%
G_max=100;                           %Maximum no. of generations%
x_min=1;                             %Minimum search range%
x_max=21;                            %Maximum search range%
A=xlsread('data.xlsx');
%Initialization%
for i=1:NP
   x(i,:)=randi([x_min x_max],1,D);
   while size(unique(x(i,:)),2)<D
      x(i,:)=randi([x_min x_max],1,D);
   end;
   x(i,:)=sort(x(i,:));
   f(i)= evaluate(x(i,:),A);
end;
disp(f);
maxfit(i)=max(f);
```

```
best=x(i,:);

 for G=1:G_max
    for i=1:NP

        %Mutation%

        r=randi([1 NP],1,3);
        while size(unique(r),2)<3|r(1)==i|r(2)==i|r(3)==i
            r=randi([1 NP],1,3);
        end;

        for j=1:D
            v(i,j)=floor(x(r(1),j)+F*(x(r(2),j)-x(r(3),j)))+1;
            if v(i,j)<x_min
                v(i,j)=x_min;
            end;
            if v(i,j)>x_max
                v(i,j)=x_max;
            end;
        end;
        while size(unique(x(i,:)),2)<D
            for j=1:D
                v(i,j)=floor(x(r(1),j)+F*(x(r(2),j)-x(r(3),j)))+1;
                if v(i,j)<x_min
                    v(i,j)=x_min;
                end;
```

```
        if v(i,j)>x_max
            v(i,j)=x_max;
        end;
    end;
end;

%Crossover%

for j=1:D

    if rand<CR
        u(i,j)=v(i,j);
    else
        u(i,j)=x(i,j);
    end;
end;
while size(unique(u(i,:)),2)<D
    for j=1:D
        if rand<CR
            u(i,j)=v(i,j);
        else
            u(i,j)=x(i,j);
        end;
    end;
end;

fit(i)=evaluate(u(i,:),A);
```

```
%Selection%

        if fit(i)>=f(i)
            x(i,:)=u(i,:);
            f(i)=fit(i);
            if fit(i)>=maxfit
                best=u(i,:);
                maxfit=fit(i);
            end;
        end;
    end;
end;
```

```
%Fitness Function%

function f = evaluate(x,A)


    data_prime=A(:,x);
    [nrows,ncols]=size(A);
    Y=A(:,ncols);
     N=size(A,1);
    training_data_1=A(1:37,:);
    training_data_2=A(113:150,:);
    training_data=[training_data_1;training_data_2];
    testing_data=A(38:112,:);
     training_group=training_data(:,ncols);
```

```
    testing_group=testing_data(:,ncols);
    svmstruct=fitcsvm(training_data,training_group);
    classes=predict(svmstruct,testing_data);
    cp=classperf(testing_group,classes);
    f=cp.CorrectRate;

end
```

Result:

When the above Matlab Code is executed with the above described dataset, it gives a classification accuracy of 97.23%.

3.4: Conclusion:

When Differential Evolutionary algorithm is used for feature selection of the dataset, we obtain a high classification accuracy of 97.23% . On the other hand, when the feature selection is not performed, a classification accuracy of 94.67% is obtained .

Hence from the result itself, it is clear that the classification accuracy improves when feature selection of the dataset is performed. By performing feature selection using Evolutionary optimization algorithms, the redundant features are eliminated from the dataset and the computational complexity is reduced. Since, the optimization algorithms selects only the optimal subsets and discards the irrelevant feature subsets, the classification accuracy improves significantly.

3.5: References:

[1] Saugat Bhattacharyya, Pratyusha Rakshit, Amit Konar, D.N. Tibarewala, Ramadoss Janarthanan, "Feature Selection of Motor Imagery EEG Signals Using Firefly Temporal Difference Q-Learning and Support Vector Machine." B.K. Panigrahi et al. (Eds.): SEMCCO 2013, Part II, LNCS 8298, pp. 534–545, Springer International Publishing Switzerland 2013

[2] Gunnar Ratsch. "A Brief Introduction into Machine Learning." Friedrich Miescher Laboratory of the Max Planck Society, Spemannstrabe 37, 72076 Tubingen, Germany

[3] Theodoros Evgeniou and Massimiliano Pontil "Support Vector Machines: Theory and Applications." Center for Biological and Computational Learning, and Artificial Intelligence Laboratory, MIT, E25-201, Cambridge, MA 02139, USA

[4] Jiliang Tang, Salem Alelyani and Huan Liu, "Feature Selection for Classification: A Review." National Science Foundation under Grant No, IIS-1217466

Chapter 4: Conclusion and Future Works:

4.1: Conclusion:

Optimization is everywhere, from financial markets to engineering design, from our daily activity to industrial applications. We always try to either maximize or minimize something. An organization wants to maximize its profits, minimize costs and maximize performance. However, most of the real life problems are non-linear, high-dimensional and consisting of complex search space and involving a lot of local minima and maxima. These issues get more complicated in case of large dataset. In such cases, traditional methods such as techniques of differential calculus fail to perform. Here comes the importance of stochastic methods or evolutionary algorithms. These algorithms perform a more exhaustive search of the model space.

The purpose of our study is to optimize a BCI based dataset using a standard evolutionary algorithm known as Differential Evolution and then to obtain its classification accuracy.

Chapter 1 gives a general idea about Brain Computer Interfacing. It discussed in brief the real life applications, the different steps and the different modalities of BCI. It also mentions the various advantages of using EEG-based BCI and the various features of an EEG signal. Also, the importance of feature selection and role of evolutionary optimization in EEG-based BCI are discussed.

Chapter 2 gives a brief introduction of Differential Evolutionary optimization. It contains literature review, the main steps and the pseudo code of DE. The literature review is based on the existing works in the field of DE. The chapter also mentions a few real world applications of DE.

In chapter 3, various machine learning techniques are defined very briefly. It also contains a few words on Support Vector Machine. Then the proposed methodology is discussed with a diagrammatic representation. Finally, the matlab code and the results obtained are discussed. A final conclusion is drawn from the results obtained.

Chapter 4 is the concluding chapter of the thesis. It contains mainly the summary of the entire thesis. Also, it introduces the readers to future possibilities of research in the domain of DE.

4.2: Future Scope of Research:

Although since the last two decades, there has been extensible research work with DE, there still exist some interesting open problems and new application areas that are yet to be explored. Some of the prominent future research directions in the field of DE are discussed in brief below-

- The performance of DE on computationally expensive problems is not satisfactory. Thus, new strategies are to be employed to deal with expensive problems more competitively by using DE.
- Rotation invariance is a challenge for DE. Although, covariance matrix based mutation and cross-over operations have been integrated in DE, this process suffers from the lack of scalability. It is because it involves repeated inversion of matrix whose dimensionality is the same as the dimensionality of the problem. These challenges open a future research direction with DE.
- The convergence speed of DE is slow because of the randomized mutation operation and parent-offspring competition. However, Euclidean distance based neighborhood can be investigated in the context of improving the convergence issue of DE.
- Learning based approaches for DE are yet to be investigated. This approach is likely to solve minor variants of the same problem repeatedly by DE within short period of time.
- In case of population based algorithms, one interesting research direction may be controlling diversity and convergence behavior while avoiding chaotic search behavior within the given computational budget.
- The convergence rate, EFHT (the average time required by an Evolutionary algorithm to find an optimal solution for the first time) are the measures of

average computational complexity of any optimization technique. The theoretical investigation of these parameters is yet to be developed.

- In case of solving real world optimization problems, there is lack of a clear mapping between problem features and the best suited optimization algorithm (i.e., out of DE, GA, PSO, ABC etc.). Not much attention has been given on the study to determine which problem features what kind of correlation among a set of decision variables make an objective function solvable by DE.
- Many methods introduce certain amount of randomness while adapting control parameters like Cr and F. There is a potential research future in exploring whether this randomness can be controlled (i.e., increased or decreased) to suit some features of the function to be optimized or on some kind of correlation among the decision variables.

Hence, there is an enormous future scope of research in the field of optimization using DE. To add to it, these research areas can be investigated collectively as well as we can treat them in different optimization scenarios (such as dynamic, constrained, multi-modal etc.).