Electroencephalography Based Brain Controlled Grasp and Lift

by

Pritam Chakraborty
18101357
Sanjida Akter
18101215
Bariyat Hasin
17101300
Syeda Faria Ahmed
18101161

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
September 2021

Declaration

It is hereby declared that

1. The paper we submitted is the result of our own unique research, which we conducted while studying at Brac University.

2. The study does not incorporate anything previously published or created by a third party unless it is properly referenced by complete and correct referencing.

3. This paper does not contain any materials that have been accepted or applied for a degree or certificate from any academic or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Pritam Chakraborty

Pritam Chakraborty
18101357

Sanjida

Sanjida Akter
18101215

Bariyat

Bariyat Hasin
17101300

Ahmed

Syeda Faria Ahmed
18101161

Approval

The thesis titled "Electroencephalography Based Brain Controlled Grasp and Lift" submitted by

1. Pritam Chakraborty (18101357)
2. Sanjida Akter (18101215)
3. Bariyat Hasin (17101300)
4. Syeda Faria Ahmed (18101161)

Of Summer, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on September 26, 2021.

Examining Committee:

Supervisor:
(Member)

Zavid Parvez
Digitally signed by Zavid Parvez
DN: cn=Zavid Parvez, o=Brac University, ou=CSE, email=zavid.parvez@bracu.ac.bd, c=US
Date: 2021.10.05 22:33:17 +11'00'

Dr. Mohammad Zavid Parvez
Associate Professor
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

RAFEED RAHMAN

Rafeed Rahman
Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Assistant Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi
Associate Professor
Department of Computer Science and Engineering
Brac University

Ethics Statement

Our thesis is of the utmost caliber and is entirely reliable. The confidentiality and integrity of our thesis paper participants are extremely important to us. Our analysis is independent and impartial since we conducted our thesis in an unbiased manner. Hopefully, this thesis analysis will add some additional features to humankind's progress in the future.

Abstract

The complexities and challenges of performing daily fundamental activities like getting dressed, answering a phone call, opening and closing the door, writing something down, or even consuming foods for patients who have lost their functionality of hands and arms due to neurological disability or amputations, is something anyone could never imagine. Our research paper serves to show the importance of restoring patients' ability to do daily activities to increase their mobility and standard of living. In this paper, we have proposed an innovative, resilient and dynamic implementation of a grasp and lift technology that would accumulate brain signals in the form of waves to operate prosthetic limbs without the help of an external device and wires. We have decided to use Electroencephalography to reactivate the neuromuscular bypass with the help of an EEG device for obtaining brain signals that correspond to specific circumstances from the scalp surface area. We also have established models using Neural Networks that would monitor multimodal sensory activities which include object encounter, grasp, lift-off, replacement from the dataset and assist the users of this technology to operate the prostheses only by incorporating their brain signals. The artistry of the whole procedure incorporates substantial segments like signal acquisition and pre-processing of the signals into data, feature extraction, denoising etc. which later leads us to implement CNN and LSTM models. After implementing the models we obtained the accuracy of 90.11% and 74.44% from the CNN and LSTM model respectively. Throughout the implementation, there will be a differential boost in the accuracy level for each of the models. Therefore, our paper is an evidence of how EEG is considered to be a communication channel between prosthetic devices and the human brain. Furthermore, it intricately reveals the approach of grasp and lift technology through signal acquisition, processing, and implementation based on Electroencephalography.

Keywords: BCI; Electroencephalography; Brain Wave; Memory Cell; Signal Processing

Acknowledgement

First and foremost, we express gratitude to the Almighty for keeping us safe and sound during this pandemic crisis and allowing us to continue our study work and put up our best efforts to successfully complete it without serious difficulties. Second, we would like to express our sincere gratitude and respect for Dr. Mohammad Zavid Parvez, our esteemed supervisor, for his relentless efforts, unwavering encouragement, and continuous support throughout the research. Third, we are extremely fortunate and grateful to have Mr. Rafeed Rahman as our co-supervisor. Furthermore, it may not be achievable without our parents' constant support. Thanks to their tremendous encouragement and prayers, we are currently on the verge of graduating. Last but not least, we are grateful to Brac University for allowing us to carry out this research.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

ϵ Epsilon

υ Upsilon

CWT Continuous Wavelet Transform

DWT Discrete Wavelet Transform

FFT Fast Fourier Transform

KNN K-Nearest Neighbours

MLP Multilayer Perceptron

$ReLU$ Rectified Linear Units

SVM Support Vector Machines

XGB Extreme Gradient Boosting

Chapter 1

Introduction

The human brain is the command center for the human nervous system that contains one hundred billion neurons, which are connected by trillions of synapses. These interconnected neurons receive signals from the body's sensory organs and generate various neuronal activity patterns that correspond to different emotions and thoughts. The patterns of neural interactions between the brain cells depends on different and frequently changing states of the human brain. Likewise, when the human brain is not functioning it generates Alpha waves. However, when the human brain is in an active state, it creates four different types of waves - Beta, Theta, Gamma and Mu.

A brain computer interface (BCI) is a computer-based system that evaluates these signals and converts them into instructions that are then sent to an output device to perform the desired activity. Electroencephalogram (EEG) is one of the most commonly used approaches for BCI due to the convenience and non-invasive implementation. Noninvasive BCIs collect the signals from the scalp and earlobe by using electrodes which consist of small metal discs with thin wires. As EEG is not invasive in nature, it is not hazardous to health as well as cost effective.

EEG, a technique based on Brain Computer Interface (BCI), has offered a means of grasp and lift by controlling prostheses for patients who are unable to operate certain limbs due to physical limitations considering the fact that one of the biggest challenges for physically disabled people is mobility. In recent times many technological devices or systems have been introduced that help the physically disabled people specifically. However, there are some cases where patients are unable to use the modified wheelchair, joystick or tactile screen with voice recognition system as they might suffer cases of dysmorphia or amputation or unable to use certain functions of that technology due to physical limitations. The grasp and lift technology has been developed to the extent that the lifestyle of the physically restrained people has come to be at ease. The grasp and lift technology can operate just by collecting the brain waves of the users through the electrodes, even without any help of other external substances.

Neural network is a widely used term for the data processing of EEG signals. Deep learning is a type of machine learning in which the algorithms are based on the human brain's structure. Deep learning is built on the foundation of neural networks. In daily life neural networks are used in different fields such as image recognition, predict scores, medical diagnosis, intelligent searching, voice recognition etc. Neural network is an artificial and modified form of Biological Neural Network (BNN) which is also present in our study too. In this research we are improvising the accuracy of moving our body parts through EEG signals. To process the data of EEG signals there are few types of neural networks which are Artificial Neural Network (ANN), Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). Long short-term memory (LSTM) is an artificial rnn - based (recurrent neural network) architecture that replaces these sorts of neural networks. These neural networks are vastly used to preprocess the data and acknowledge the pattern to predict accuracy. We can use several neural networks in the system of grasp and lift but the main goal is to find the accurate one. Here the main purpose is to assist humans to increase their mobility and develop a helpful system to help them with amenity.

Every year a huge number of people's disabilities happen because of road accidents. Those who suffer from disabilities perform their daily tasks. Almost 90% of the disable persons are from road accident and others suffers from various diseases. Another important term Neuromuscular disorder is a disease for disabilities. Around 1 to 10 in per 100,000 population suffers from this disease. Focusing on helping disable persons to restore their abilities to perform daily tasks EEG based grasp and lift will be a future for disable persons. The grasp and lift technology developed with the help of EEG will bring an extremely high impact, as the physically impaired individuals will be able to operate their upper limbs by themselves without any additional devices or wires attached around the body. The movements of the prostheses will be solely configured during motorizing the fingers and moving parts by comprehending and replicating the ways humans perform grasping with fingers and palm and lifting the upper limb by the brain waves that was inferred by the EEG method previously. It can also be said that it is very convenient to use for most of the users. The components used in this grasp and lift technology are also very low cost yet have an optimum performance level. It makes use of upcoming and ever evolving technology that will make iterations simple and manageable.

1.1 Research Problems

Every year a lot of people suffer from disabilities which occur genetically or by accidents. This force those people to rely more on devices to continue daily life. Keeping this in mind, the idea of a Brain controlled grasp and lift technology was introduced which can reduce the hassle of the disabled people with amputations or dysmorphic limbs. Prosthetic limb is a commonly used thing for physically disabled people to give portability to their daily life. However, most of the prostheses need numerous attachments and devices connected to their body to operate. But EEG based grasp and lift allows the user to operate the limb only by using brain signals. So, for those people who have lost their ability to move, do not have to depend on anyone and can easily lift and move their upper limb around using this device.

The life of a disabled and a paralytic patient is far more difficult than anyone can imagine. People who have lost their limbs in accidents or amputations or have any sort of dysmorphic limbs are among the most visible members of the community and they experience a very high level of physical disabilities. Though it is easily said that prosthetic limbs are actually helping the disabled and the paralytic patient, there are certain cases of major upper limb mobility issues and even greater degree of amputations and disability where the patient might not be able to operate the limb by themselves considering the numerous number of wires attached to their body in order to operate an arm. Even in the cases of other technologies like joystick, tactile screen and system based on voice recognition or wheelchair integrated with ultrasonic sensor and blinking sensor, there are numerous cases where the patients are unable to use these technologies for a number of reasons.

The activity of the brain is necessary for implication of BCI for grasp and lift technology. However along with some advantages, there are some drawbacks in this system. BCI techniques are categorized in two sectors which are invasive and non-invasive. In an invasive system the electrodes are placed directly into the brain and connected to the computer itself. Invasive method shows higher performance than the other but it is also dangerous for health as it is directly connected to the brain. As well as, it collects data signals from the brain and translates the algorithm to the device. This invasive method can create scar tissue in the brain. But as we are using EEG devices to collect brain signals from the scalp and translate it to the application through Arduino. Moreover, non-invasive methods are more secure than invasive methods as the electrodes are placed outside and it is more compact and easy to use for disabled people. Furthermore, there are different levels of paralysis such as Monoplegia, Hemiplegia, Paraplegia and Quadriplegia. So, the people with Hemiplegia cannot go through this process as this system is operated by brain signals.

Regardless of the meaning of signals, we live in such a time and space where thousands of other waves or signals are around us. So, it is obvious to get mixed up with these signals as an EEG device generates electrical waves to operate the system. Five kinds of waves such as Alpha, Beta, Theta, Gamma, Mu which are produced by the brain on different occasions. So, the BCI technology is crude. But this EEG device is sustainable for future purposes. Using the above mentioned technique, a simple EEG controlled grasp and lift has been proposed in this paper.

1.2 Research Objectives

This research aims to develop a suitable platform for human beings to perform tasks of grasp and lift by using brain waves capturing it by EEG device and modifying it. The sectors we are focusing through our researches are -

1. In this paper we focused on preprocessing part so that we can have better signal to noise ratio to minimize the amount of redundant data. We have used common average reference (CAR). Moreover, we collected extensive number of data to have better to have better result.

2. In this paper we have executed multiple neural network models to compare

the accuracies and to find the precised result for the system. We prepared our models to find the finest model for grasp and lift. Moreover, in this paper we visualized the models to have a better understanding of the models. This paper is expected to be a good resource for the future work in the sector of Grasp and Lift technology.

Chapter 2

Related Works

Before progressing with this research, we had gone through some of the research papers related to this field to gain better perspective and understanding about the concepts involving our research. We have come across other research papers that remarkably helped us through our research process. There has been a significant number of thesis works and papers based on Electroencephalography and different arenas of EEG and its methodologies. The major goal of our paper is to establish a Grasp and Lift technology with EEG and Neural Network that could be useful for people who are suffering from upper body mobility issues due to neuromuscular disorders and amputations. Our task starts with extracting brain signals as data and utilizing them to control a certain prosthetic limb or upper limb exoskeleton without any execution of external devices. We have proposed models using Neural Network to supervise multimodal sensory activities like hand movement, grasp, lift-off, encounter, object replacement etc providing us a differential boost in the accuracy level of each model.

This section indicates the importance of other research and how the works have impacted our process of understanding. We have critically analyzed the concept of BCI, EEG, Neural Network and each distinguishes methods that have been implemented in constructing brain-controlled Grasp and Lift technology from the paper works by other researchers. We have also discussed the importance of this technology in the life of a person with upper body mobility issues. The research works we have reviewed and analyzed also gave us a complete idea of the process of collecting brain waves through EEG devices and implementing EEG based grasp and lift with the assistance of neural networks for real life use.

According to the paper [15], the authors had shown the prediction of movement of hands while performing grasp and lift tasks from EEG signals. They focused on Recurrent Neural Networks by varying their architectures and then noted their outcome and performance. They had shown the use of variant architecture which were MUT1, MUT2, and MUT3. The classification accuracy shown by MUT1, MUT2, and MUT3 were 86.54%, 88.21% and 88.82% respectively.

The author of the paper [7] had shown a Fuzzy logic based approach of a robotic system which can grasp and lift objects and also has the capability to handle fragile objects. The fuzzy logic is used to control the speed of movement of fingers which is based upon clustering. They gathered initial I/O data based upon successful grasp and lift attempts which were used for the purpose of training the fuzzy system.

A recent research work by the researchers of Multimedia University, Malaysia [18] has proposed a development of a brain-controlled wheelchair system. The research has shown the results of different BCI methods applied in human brain and results shows that EEG has been proven to be the most suitable method to utilize for retrieving the signal of brain activity in order to implement the BCI system than Single Photon Emission Tomography, Positron Emission Tomography and Functional Magnetic Resonance Imaging etc. It also mentions different states of brain activity resulting in different types of brain waves. It has also introduced a unique footprint ID that catches into the sensor and creates vibration for different activities.

The research work [27] in "The Brief History of Brain Computer Interfaces" represents the emergence of Brain Computer Interfaces and also leaves a hint of the future of BCI technology. The article briefly describes the time period of the 1970s, when BCI research and development solely focused on neuro-prosthetic application to restore damaged eyesight, hearing and movement. That time Hans Berger was inspired by Richard Canton, who discovered electric signals in animals' brains; and his innovation in the field of human brain research and its electrical activity has a close connection with the discovery of brain computer interfaces. Later in 1998, researcher Philip Kennedy successfully implanted the first BCI object into a human being. And from then, BCI has introduced numerous methods and systems that have enriched this field.

From the discussion mentioned above, the impact of previous works reflected in our research is clearly noticeable. Our paper has described the elaborative concept of BCI, invasive and non-invasive methods of attaining the brain signals, different states of the human brain that releases different types of brain waves reacting to circumstances and environment. implementation of EEG on prosthetic arms. For this scenario, it is also observed why EEG is considered to be the best suitable method for grasp and lift.

The paper "Brain-computer interface technology: A review of the first international meeting" stated BCI [4] as a 'wire-tapping' analogy which doesn't represent the development of BCI. The main focus of this study was to describe the first international meeting that was held in New York in June 1999. The findings of 22 different groups stated the research methods and the future perspective of BCI. The paper demonstrated the two adaptive controllers of BCI where one was said to be users' brain which fabricates the activity which is measured by the BCI system and another one is the system itself which extracts those activity in a certain command. The proposed paper exhibits some invasive methods for the BCI signals which describes the possible location for implemented electrodes, improvement of recording BCI technologies to development in the future. FMRI and MEG were being discussed as one of the significant technologies to develop the field of recording the BCI. Applications of BCI technology were also being described in the paper.

In [15], the authors have proposed a system that identifies hand motions during a grasp-and-lift task from EEG recordings by using RNN. Their proposed system was competent to detect the hand motions during the movements. In their system, they have used RNN and the architectures of RNN like LSTM, GRU, and its variants named MUT1, MUT2, and MUT3. In comparison to all the models, MUT3 manifests the best performance. To retain the consistency of the predictions and eliminate the incongruous prediction errors, they have smoothened the prediction while training the dataset in their model.

In the research work [14] of "Extracting and Selecting Distinctive EEG Features For Efficient Epileptic Seizure Prediction", the authors have presented extensive feature representations for the EEG signal to attain a comprehensible prediction of the epileptic seizures. For epilepsy patients and those who care for them, precise and consistent predictions of imminent seizures are crucial. For effective seizure estimation and forecasting, distinct seizure-related representations from EEG signals are favorably advisable. Furthermore, they have implemented a parameter screening procedure on the amplitude and frequency components to minimize possible redundancies. They also mentioned that because of the possibility of personal variation, the feature selection procedure has been performed on a patient-specific basis. Moreover, according to their research, their methodology surpasses state-of-the-art approaches in numerous areas, including prediction sensitivity, prediction specificity, and successful patient rates, when compared to state-of-the-art methods under similar setup conditions dataset in their model.

Another research work by University of California [23] introduces the upper limb exoskeleton system. There are a number of cases where a prosthetic arm is unable to serve the purpose of assisting people with neuromuscular disorder, here patients might not need a replacement limb rather little bit of support for upper body movement. Exoskeletons are wearable devices that are placed into a person's body and that are useful for strengthening and regaining a patient's upper body movement. In the research paper, a sketch was mentioned that was used to implement an upper limb exoskeleton. According to the paper, it was precisely stated that for sensing methods, Electroencephalography (EEG) was a better option than Electromyography (EMG). The main reason EEG is better is because of the process of extracting the signals from the human brain and the execution of exoskeletons. In EMG, small electrodes are inserted through the skin into the muscle in order to measure the neuromuscular response and brain impulses responding to certain activity or emotion. In this case, every time a new device needs to be built for a new patient. On the contrary, EEG doesn't require to collect brain impulses from that particular person, nor does it require to do it every time a new patient comes up. Electrodes are placed in a EEG cap that particularly collects the brain waves from the scalp and behind the earlobes and hence all the collected signals are programmed into dataset that are used to build a universal exoskeleton system that works for everyone. Moreover, this paper provides a cost friendly solution of coping up with neuromuscular disorder by using EEG.

Chapter 3

Background Studies

3.1 Brain Computer Interface (BCI)

Brain Computer Interface is a technology that analyzes the signals generated by brain activities and converts them into control signals to control any kinds of external devices without the intervention of peripheral nerves and muscles. BCI technology enables its users to interact with computers and computer-based systems by means of brain activity only. Specifically, through brain waves. The concept of BCI is not new, in fact, the emergence of BCI has been traced back to the 1970s. A renowned article [27] named "The Brief History of Brain Computer Interfaces' ' clearly states that research on BCI started at the University of California in the 1970s, which led to the emergence of the expression Brain Computer Interface. But in the mid-1990s, BCI added a whole new dimension in this field by marking the appearance of the first neuro-prosthetic devices for humans. Information has also been found in research that BCI cannot read the brain waves accurately, it recognizes specific frequency patterns of the brain and also can detect changes in the energy when humans think or act in a certain way. The purpose of BCI is not only to detect brain activity and predict his next work too. BCI can be described as a loop of controlling the whole system of the process. Controlling the brain signals and pre-processing it to a specific data that can be read through by machines and performed is the overall process of BCI signals.

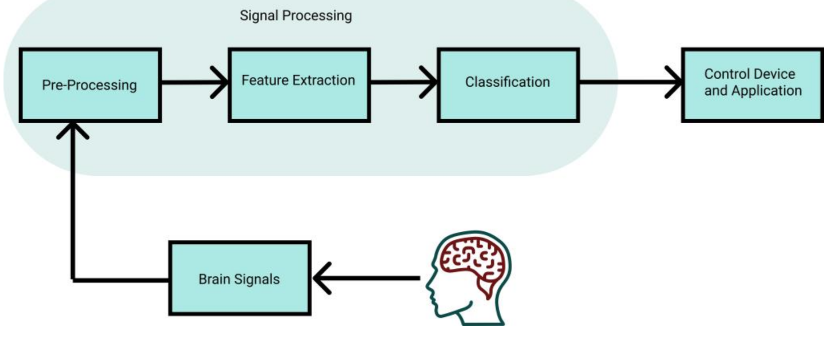

Figure 3.1: Brain Computer Interface

3.2 Electroencephalography (EEG)

According to [9], Electroencephalography is an electrophysiological monitoring technique where the activity of the brain is recorded and measured through the device, Electroencephalogram (EEG). This device consists of electrodes, conductive gel, amplifiers and Analog to digital converter. The electrodes are embedded in a cap and the resistance of the wires must be less than 5 ohms. To interpret the collected brain waves, the position of the electrodes has to be in the exact position. Furthermore, the electrodes are labeled by the first letters of the lobes of human brains and the odd number indicates the left hemisphere whereas the even number represents the right hemisphere. Additionally, these electrodes measure some special applications such as heart rate, eye movement, skin conductance and respiration etc. The collected signals are being processed in the processing phase. For better prediction the proper result of EEG is needed.

The most clinically researched method name is 10-20 system which is an international system for electrode location. 19 electrodes to record signals in most applications [6]. But in some case 256 electrodes are also used in the scalp. So, in case of displaying the wave two types of filter are being used one is high pass filter and another is low pass filter.

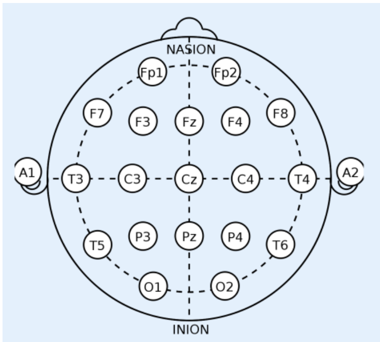

Figure 3.2: Electrode Location System

The brain wave which comes from Scalp is very small that minimum voltage is required to measure. Alpha, Beta, Gamma, Theta, Delta, Mu have different kinds of frequencies that are captured by EEG devices. There are several types of EEG portable headset nowadays. EEG headset is a wearable device that monitors and records brain activity and applies it to different applications. There are basically two types of EEG headset one is Commercial EEG headset and another one is Open Source EEG Headset. Fast Fourier System, Discrete Wavelet Transformation and Entropy based feature extraction systems are the most commonly used.

3.3 Fast Fourier Transform (FFT)

This method mathematically determines the form of EEG data signals. Power Spectral Density (PSD) computes the acquired data signals from EEG devices and shows the EEG data samples. This PSD method is calculated by the Fourier method. The Welch method is one of the traditional methods of PSD. The sequence of *xi(z)* can be described as

$$x_i(z) = x(z + iD), \quad z = 0, 1, 2, \ldots, M - 1$$
$$\text{while } i = 0, 1, 2, \ldots, L - 1;$$

Take iD as the starting point of the sequence ith. Data segments that are formed are represented by L of length $2M$. The output periodogram is

$$P_{xx}^{\approx(i)}(f) = \frac{1}{MU} \left| \sum_{z=0}^{M-1} x_i(z) w(z) e^{-j2\pi fz} \right|^2$$

In this window function the term U hows the normalisation factor of the power and it represents in such a way that

$$U = \frac{1}{M} \sum_{z=0}^{M-1} w^2(z)$$

w(z) is a windowed function. Average if periodograms show us the Welch's spectrum which is given below

$$P_{xx}^{W} = \frac{1}{L} \sum_{i=0}^{L-1} P_{xx}^{\approx(i)}(f)$$

3.4 Wavelet Transform

There are few disadvantages of the Fourier Transform. If we represent the Fourier Transform in the time axis we will not be able to determine in what moment a specific frequency will go up [17]. Though, Short-Time Fourier Transform (STFT) uses a window to display the change of frequencies over time but the window has limitations. So, to overcome the situation wavelet transform is used. Haar Wavelet in 1909 proposed the idea of Wavelet. Though the concept of wavelet didn't exist at that time, that's why Jean Morlet in 1981 proposed the concept. In 1984 Morlet and Alex Grossman discovered the term wavelet. Wavelet analysis can be done in two ways one is Continuous Wavelet Transform (CWT) and another one is Discrete Wavelet Transform (DWT).

3.4.1 Continuous Wavelet Transform

The computation of input data sequence and a set of functions produced by the mother wavelet is called Continuous Wavelet transform. For our signal f(t) the equation is

$$CWT(x, y) = (f, \Psi_{x,y}) = \frac{1}{\sqrt{x}} \int_{-\infty}^{\infty} f(t).\Psi\left(\frac{t-y}{a}\right) dt$$

Here, x and y are two parameters which are actually the scale and position. Scale parameter is responsible for compressing the wavelet or stretching the wavelet and for y which is position parameter the wavelet function extends around the t axis. All the coefficients are calculated for the different values of x. CWT are mainly used for spectral analysis of signals. It can be used for study frequency break, time discontinuity, Signal burst etc.

3.4.2 Discrete Wavelet Transform

From these two types DWT is the most commonly used for feature extraction. There are few non-stationary and temporary characteristics which can't be done by Fourier transform but the DWT method can be used by extracting EEG signals as the time frequency has the advantages of localization, multirate filtering and scale space. The signals go through the Low Pass Filter and High Pass filter.

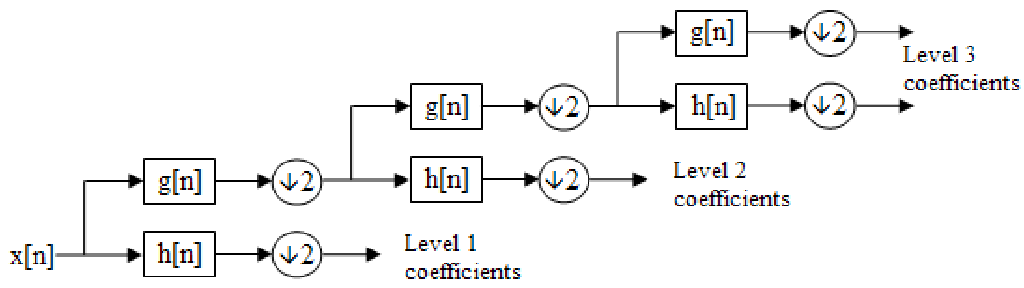

Figure 3.3: Discrete Wavelet Transform

3.5 Autoregressive Model

The Autoregression (AR) model is one of the broadly used models in EEG. Autoregression model is used for signal recognition and processing. In AR the variable prognosticates by linear combination of previous values. The coefficients of the AR model are used as feature vectors in BCI [10]. AR model is the linear regression of the current observation of the series. The equation that can represent the AR model is

$$A_t = c + \sum_{i=1}^{p} \varphi_i A_{t-i} + \varepsilon_t$$

Here A_t is known as the time series and ϕ_i are the parameters and c is constant.

After the extraction of the features of the signals, we have to classify them. Some neural networks have been used to achieve our proposition like as Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), K-nearest Neighbors (KNN), Multilayer Perceptron (MLP) and Adaptive Boosting (AdaBoost) etc. All the models have performed strongly in regards to correct classification. The broad discussion has been made following sections.

3.6 Classification Algorithm

3.6.1 KNN

K-Nearest Neighbour known as KNN is a Machine Learning Algorithm constructed by the Supervised Learning Technique. It is mainly applied for classification and regression. This algorithm categorizes a new data point that is similar to the available data. Moreover, it imputes the strayed values and resamples the datasets.

There are three types of learning of this algorithm - Instance-based learning, Lazy learning, Non-parametric learning. In Instance-based learning, rather than learning weights from the training data in order to estimate the output, the entire training instance is used to anticipate the output for unknown data. Furthermore, in Lazy-learning, the learning process is deferred when a prediction is required on the new instance, and the model is not learned using the training data beforehand. In Non-parametric learning of KNN, the mapping function does not have any specified form and does not construct any presumptions about the underlying data.

In addition, during the training process, the KNN algorithm basically preserves the dataset, and when new data is received, it tries to find the nearest neighbor of the point. The data points with the shortest distance in feature space from the new data point are the nearest neighbors. Additionally, the K of KNN represents the number of such data points that have been taken in account while implementing the algorithm. Therefore, while applying the algorithm, the value of K and the distance have to be evaluated correctly. The distance metrics that can be used in this algorithm are Euclidean distance which is mostly used, Hamming distance, Manhattan distance, and Minkowski distance.

At first, the algorithm chooses the number K of the neighbors. After that, in most cases, it evaluates the distance metric by using Euclidean distance of K number of neighbours. The K nearest neighbour is selected according to the calculated distance. Next, among these neighbours the number of the datasets is counted for each category. Lastly, a new data point is assigned to the category with the greatest number of neighbours.

The KNN algorithm is simple to put into action, can withstand noisy training data and if the training data is vast, it can work efficiently. However, it must consistently evaluate the value of K, that can be complicated at times and as the distance between the data points for all of the training samples must be calculated, the calculation cost gets high [13].

3.6.2 SVM

SVMs, or Support Vector Machines, are supervised learning models that are extensively used to solve classification and regression concerns. It works with both linear and non-linear systems and can be used in a variety of situations. SVM works on the basis of developing a line or hyperplane that splits data into classes. A hyperplane is a subspace that has one fewer dimension than the surrounding space, hence the hyperplanes of a three-dimensional space are the two-dimensional planes. Support vectors are also the coordinates of each individual observation. It is this line that most effectively divides the two categories.

In the SVM classifier, constructing a linear hyper-plane between these two classes is trivial. Another notion applied in the SVM method is the kernel trick. The SVM kernel is a function that uses a low-dimensional input space and transforms it to a higher-dimensional space to convert a non-separable problem into a separable one. It's very useful when dealing with non-linear separation issues. It goes through a series of highly complex data transformations before deciding how to split the data using the labels or outputs specified.

3.6.3 MLP

A multilayer perceptron (MLP) is a feedforward learning type of Artificial Neural Network (ANN). An MLP has at least three layers of cells: the input layer, the hidden layer, and the output layer. Except for the input nodes, each node is a neuron with a nonlinear activation function. MLP implements backpropagation, a supervised learning approach, throughout its training [2]. Unlike the other different classifiers such as Support Vectors or Naive Bayes Classifier, the MLP classifier completes the classification task using an underlying Neural Network [24].

Furthermore, the MLP drives the inputs forward by calculating the dot product of the input with the weights that reside between the input layer and the hidden layer (WH). This dot product generates a value at the hidden layer. MLPs utilize activation functions at each of their calculated layers after that. Only a few of the activation functions to consider are Rectified linear units (ReLU), sigmoid function, and tanh. To push the calculated output to the current layer, apply any of these activation functions. Additionally, it also feeds the derived output from the hidden layer to the MLP's next layer by taking the dot product with the corresponding weights after it has been driven through the activation function. Steps two and three should be performed until the output layer is reached. In the output layer, the calculations will be used for either a backpropagation approach that corresponds to the activation function chosen for the MLP (in the case of training) or a choice based on the output (in the case of testing). MLPs are the cornerstones of all neural networks, and when utilized to solve classification and regression problems, they have significantly increased computer computational capability [25].

3.6.4 K fold Cross Validation

Cross-validation, also known as rotational prediction or out-of-sample testing, is a model validation methodology for establishing how well a statistical study's results would generalize to a new set of data. It's most frequently used when the user wants to know how well a prognostic model would work in practice.

In most cases, a model is provided with both a dataset of known data to train on and a dataset of unknown data to test against. Cross-validation is a technique for evaluating a model's potential to anticipate new data that was not used in its prediction. It is used to identify the problems like overfitting and selection bias, as well as to provide insight into how the model will generalize to a different dataset. Therefore, cross-validation integrates (averages) fitness metrics in prediction to provide a more accurate estimation of model prediction performance [21].

When cross-validation is utilized for both determining the optimal set of hyperparameters and estimating error(and assessing generalization capability), layered cross-validation is required. There are a few different versions. K*I-fold cross-validation and K-fold cross-validation with validation and test set are two separate variations. The K-Fold CV method divides a data set into a K number of segments, each of which functions as a testing set at some point. As a result, k refers to the data set's subsets. Each time, one of the k subsets is chosen as the test set, with the remaining k-1 subsets forming a training set. After that, the average error through all k trials is computed.

3.6.5 Random Forest

Leo Breiman in 2001 proposed the Random forest algorithm. Random forest is a supervised learning algorithm [16]. Random Forest is a generalized form of some decision trees that predict the more accurate result. Random forest is usually trained by the data in the 'bagging' method. Random is a combination of multiple decision trees. The final decision is made by the output of the maximum number of decision trees. Random forest introduces flexibility and high variance factor into low variance. In the case of regression, it is the average of all the outputs.

Bootstrapping is one of the important methods of random forest. Bootstrapping is a method that makes our whole dataset a small set. Using bootstrapping and creating the uncorrelated models and then aggregating their result is called bootstrap aggregating or bagging. By shuffling which feature each tree can split on is known as feature randomness.

$$x = \text{is the values of random prediction}$$
$$N = \text{is the number of trees}$$
$$f_{\text{avg}}(x) = \text{Average of the predictions}$$

$$f_{\text{avg}}(x) = \frac{1}{N}\sum^{N} f(x)$$

3.6.6 AdaBoost Classifier

Adaptive boosting or AdaBoost was proposed by Yoav Freund and Robert Schapire. It is mainly used to boost any machine learning algorithm. This boosts the weak classifier into the stronger one by training weighted data. If we consider a dataset and a base learner in which some of the records from our dataset will pass through the base learner sequentially. The base learner can be any model. If any data is incorrectly classified to base learner 1 it will be passed to base learner 2 simultaneously. It will process like this until the last base learner creates and this is the way of boosting technique. In Adaboost it will be given some sample weight first which will follow the formula

$$\text{weight } = \frac{1}{n}$$

Where n is the number of records. All the records will be assigned the same weight. Next, the base learner which is some decision trees. This is different from the random forest. The decision trees will be created with the help of one depth which is known as Stumps. This will have one depth and can carry multiple leaf nodes. The features will be divided into stumps. To select the particular base learner model there is another term known as entropy. If the entropy is less for particular stumps then that stump will be selected. Next, we need to find the total error, then we need to calculate how that model performed. The error weight will be updated which is normalized weight and this process will be continuing until it passes all the stumps. Finally, we will consider that the normalized weight is less or the error is less than the error we had in the initial state.

3.6.7 XGB Classifier

XGBoost or 'Extreme Gradient Boosting' is one of the most renowned methods in machine learning. It is a supervised model which uses training data to predict a specific variable. Tianqi Chen was the person who first came up with this. XGboost is a frame that can be run in multiple languages such as C++, Java, Python. It is portable on different platforms like Mac, Linux. It is integrable in various systems. The speed is very fast in processing data and performs well that's why it is one of the methods used in machine learning. Parallelization is one of the aspects to process the data fast. It uses all the cores of the processor as well as the maximum available computation power of the system. XGBoost keeps all the intermediate information in cache so that it can perform fast. Out of memory computation is another important part of fast performance. Regularization parameters will prevent the model from overfitting which increases the performance of XGBoost.

3.7 Brain Waves

Human brain is a crucial organ that assembles sensory information from all over the body and regulates neuromuscular movements. Synapses, which are known to be the interconnections between the brain cells known as neurons, enables electric impulses to be transmitted through the neurons and proceeds to perform the action triggered by that particular brain impulse. So, brain impulses or commonly used term Brain Waves are generated by synchronised electric pulses from the billions of neurons interacting with each other. In the last few decades, there was a growing interest towards implementing the functionalities of the human brain into machines and devices. Scientists were able to introduce such a technique named Electroencephalography (EEG) based on Brain Computer Interface (BCI), a computer based system that analyzes these brain waves and converts them into instructions by computer algorithms that are related to an output device to conduct the expected action. As previously mentioned, Electroencephalography is performed through a device called Electroencephalogram that measures the frequency of the brain waves from the scalp and each of the frequencies is assigned for responding in a certain way. These brain waves are strongly associated with our activities and emotions also, for instance, when brain waves with a smaller frequency are dominant we may feel drowsy, sluggish and exhausted. Similarly, prominent higher frequencies are responsible for our charged, high energy activities. Brain wave frequencies around 13Hz are considered to be used for active intelligence. It has been proved that people having learning disabilities and attention deficit disorder have paucity of 13Hz activities in their brain compromising their capacity to accomplish basic sequencing assignments and mathematical calculations. It has been recognised since the early period of EEG evolution, the provinces of a functional human brain cortex have their own intrinsic brain waves of the frequency ranging from 0.5Hz to 40 Hz. From EEG recording five main ranges of frequencies have been detected. Frequencies ranging from 0.5Hz to 4Hz are known as Delta (δ) waves, from 4Hz to 8Hz as Theta (θ) waves, 8Hz $-$ 14Hz as Alpha (α) waves, 14Hz $-$ 30Hz and over 30Hz are familiar as Beta (β) waves and Gamma (γ) waves respectively. Brain waves detecting frequencies less than 0.5Hz are commonly termed Infra-low brain waves or Slow Cortical Potentials are regarded as basic brain impulses that support higher brain functions.

3.7.1 Delta (δ) Waves

Delta (δ) waves are the slowest recorded waves of the human brain with the highest amplitude which are also concerned with predominant oscillatory activity or fundamentally familiar as Slow Wave Sleep (SWS) ranging less than 4Hz frequency. They are frequently found in infants and toddlers and people associated with the deepest level of tranquil, relaxed, therapeutic and healing sleep as well as in brain injuries, learning disabilities and severe ADHD. This state also promotes repair and rejuvenation effective for the immune system, natural healing and restoring sleep. Delta (δ) waves are responsible for annihilating consciousness and sensitivity which is why individuals with autism, ADHD and attention deficit disorder are most prominently recorded with Delta waves.

3.7.2 Theta (θ) Waves

Theta (θ) waves also have a very low frequency ranging from 4Hz to 8Hz and the changes in Theta waves are very difficult to detect with only raw EEG traces. Theta (θ) brain waves are pretty dominant while sleeping, deep meditation and spiritual awareness. These waves are also correlated with intuition, perception, instinct, fantasizing often experienced when we wake up or drift or to sleep. These are mostly associated with individuals less than 14 years old, resulting in kids that age are prone to be creative, imaginative and fantasizing about different topics that builds their instinct. Moreover, particular Theta (θ) band power increases memory capacity and successful encoding of new information in the brain. Theta (θ) waves are also detected in adults who are concerned with anxiety and behavioural inhibition.

3.7.3 Alpha (α) waves

The oscillatory Alpha (α) waves are the most consistent waves especially in the parieto-occipital region of the brain which also bridges a channel between conscious and subconscious mind. Alpha (α) waves are reported to appear on the white matter of the brain, where all the information connects with each other in order to correlate. Healthy Alpha (α) waves promote resourcefulness of the brain and coordination between active state and relaxed state contributing to accomplish the tasks needed to be done. However, these waves are considered to be the most effective waves to learn and use that learning in a competitive environment. Individuals aged more than 14 years are found with prominent Alpha (α) waves. Even though Alpha (α) ranges between 8Hz and 12Hz, oscillation between these frequencies leads to a variety of states of mind. Alpha waves peak around 10Hz, authorizing 'The power of now', the active learning and accomplishment stage. Low Alpha (α) frequencies are often associated with anxiety, insomnia and stress; on the other hand, when Alpha (α) frequencies rise up, it leads to extroversion. Furthermore, the emergence of different sub-bands of Alpha (α) waves is an indication of visual attention and contextual memory demands.

3.7.4 Beta (β) waves

Depending on the activity, the synchronization of Beta (β) frequencies is augmented for consciously obtained impulses that dominate our regular conscious state when our attention is devoted towards cognitive processes. Beta (β) waves are considered to be incorporated with fast activity, mostly dominant when our brain is engaged with integrated activities like judgement, analytical problem solving, decision making and active listening. Beta (β) brain waves have a range of 12Hz to 30Hz frequencies, which are further divided into three sub-bands: Low Beta or Beta1 (12Hz - 15Hz), Mid Beta or Beta2 (15Hz - 18Hz) and High Beta or Beta3 (18Hz - 30 Hz). Low Beta (β) state promotes a focused yet relaxed mental state, inhibited by motion. Mid Beta (β) waves alert individuals of their surroundings and selves as well as indulge them for mental ability and focus. High Beta (β) waves are responsible for hyperactive states of mind where people are more focused and alert when performing any tasks, which sometimes may lead to agitation.

3.7.5 Gamma (γ) Waves

Gamma (γ) brain waves are the fastest waves with a frequency of more than 30Hz. The small amplitude of Gamma (γ) waves stimulates simultaneous processing of tasks that are rich with information from different parts of the brain. The instant functional implementation of Gamma (γ) waves is essential to symbolize the functions involving active data processing, acknowledgement of sensory information and rapid interaction between the data. Gamma (γ) waves modulated perception and awareness to a greater degree that enhanced spiritual emergence. However, scientists discarded Gamma (γ) waves as Redundant Cerebral Clutter as it tends to become highly active, even out of the range of neural capacity, when people experience genuine love and devotion, generosity and higher moral virtues.

3.8 Machine Learning

Machine learning refers to the study of computer algorithms and the use of data to learn similarly like humans. It is a branch of artificial intelligence and it uses sample data which is known as the training data to make decisions or predictions. The process of knowing about the algorithm of machine learning can be divided into three parts [26]. First is the decision process, then an error function and finally a model optimization process. In the decision process the algorithm will basically predict the type of the data it got. Then in the error function part it checks the accuracy of the model and lastly in the model optimization part the algorithm continuously evaluates and optimizes the model until it reaches the desired accuracy. Considering the type of the feedback or signal machine learning can be classified into three basic categories.

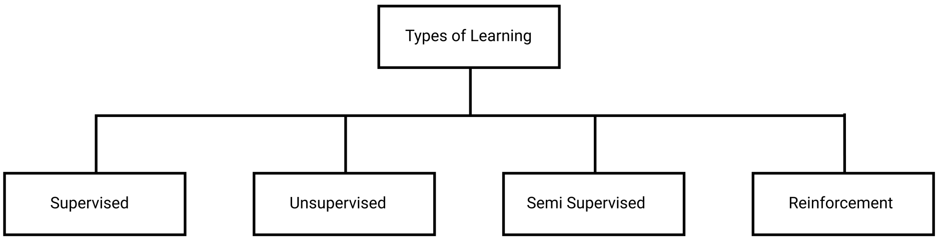

Figure 3.4: Machine Learning

1. **Supervised learning:** Supervised learning is also known as supervised machine learning. Supervised learning first builds a mathematical model of a training data set which contains inputs and outputs and these training data sets are represented using a matrix. Moreover, in the training data set there are some training examples which are represented using arrays or vectors. Each of these training examples have single or multiple inputs and an output. To determine the outputs connected to the new inputs, supervised learning learns a function through iterative optimization of an objective function. Also, for the inputs which were not in the training data set the outputs can be determined with the help of an optimal function. If an algorithm improves its accuracy of the outputs over time then it is considered to be learned. Linear regression, logistics regression, neural networks are some of the methods used in supervised learning.

2. **Unsupervised learning:** Another category of machine learning is unsupervised learning which is also known as unsupervised machine learning where it does not have any labeled training data. Unsupervised learning has a data set which only has inputs. This learning algorithm searches for similarities in data and depending on that the algorithm reacts accordingly. Moreover, this learning algorithm determines the pattern of the data without human involvement and it can also find out the similarities and differences between the data. There are some other algorithms that are used in unsupervised learning and they are probabilistic clustering methods, neural networks and so on.

3. **Semi-supervised learning:** Semi-supervised learning algorithm is the one which is in between supervised and unsupervised learning. In this learning method it has both labeled and unlabeled training data where the unlabeled training data amount is larger than labeled data. When a small amount of labeled data is combined with a large amount of unlabeled data then the learning accuracy is also improved. If there is not enough labeled data to train a supervised learning algorithm then semi-supervised learning can be a solution for this problem.

However, apart from these three categories there is another type of learning which is reinforcement learning [26]. Reinforcement learning is a bit similar to supervised learning but the difference is that the reinforcement learning algorithm does not have any sample data to be trained. This algorithm learns gradually as it moves forward by doing experiments and making errors. By using reinforcement learning a software agent can be trained to act rationally in the environment.

3.9 Logistic Regression

For prediction and statistical purposes, logistic regression has been used [5]. The coefficient of the logistical model is known as logistic regression. It is similar to linear regression, however, instead of predicting anything continuum like weight, length, so on and so forth, logistic regression anticipates something true or false. In addition, instead of fitting a straight line to the data, logistic regression uses an s-shaped logistic function. The curve starts at 0 and ends at 1. That is, we can forecast the

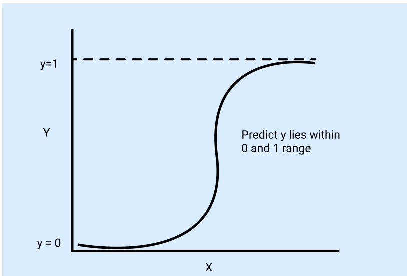

Figure 3.5: Logistic Regression

likelihood of Y based on the value of X. Logistic regression is a prominent machine learning method because of its ability to provide probabilities and classify newer samples using discrete and continuous metrics.

3.10 Neural Network

In our daily life speech recognition, face recognition, market sectors, health sector neural networks are one of the common tools. Google's suggestive search is one of the common examples of neural networks. Neural network is a working unit of deep learning. Deep learning uses neural networks to translate the activities of humans and apply it to function in different applications. Deep learning is a section of Machine Learning (ML) again where machine learning is a part of Artificial Intelligence (AI). Artificial Intelligence, Machine Learning and Deep Learning are connected where machine learning and deep learning control the process of artificial intelligence and neural networks translate data to communicate with machines.

The concept of Neural Network was proposed by Alexander Bain (1973) and William James (1890). According to Bain [1] every activity of humans is constructed through some neurons. Every time this activity repeated it strengthened the neurons. A Neural Network works when a set of data is given as input and these data go through a few layers and fabricate outputs. The input layer receives the inputs, followed by the output layer, which predicts the ultimate output. There are hidden layers in between that execute the majority of the computation required by our network.

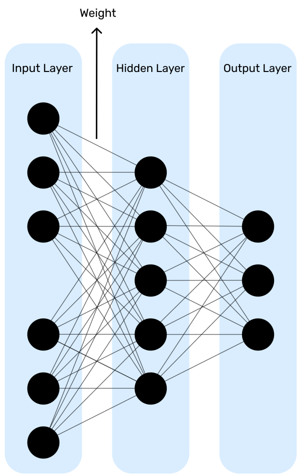

Figure 3.6: Layers of Neural Network

Here, layers are categorized into three parts: input layer, hidden layer and output layer. In Fig We can see weights. Like our brains have neurons that help to build and connect thoughts, neural networks also have neurons that get the inputs through the input layer and pass it to the output layer.

3.10.1 Artificial Neural Network

To solve the problem of speech recognition, translating text, spell checker, text classification etc ANN is a widely used field. As we have seen previously a neural network has 3 layers. Where the input layer accepts the input, a hidden layer does the process and the output layer prepares the output. The main advantage of ANN is it can take and learn any weight from any input to output. When the inputs are passed to the hidden layer an initial weight is assigned to each input. The inputs are multiplied by the weight and sum of the multiplication process through the layers to the activation function which plays a vital role in ANN. Then it goes forward until it reaches the output layer. At the output layer a probability determines the location of data. It is well known as Feed Forward Network. ANN has distributed memories.

With some advantages ANN also has some disadvantages. When we use ANN for image classification the main part of the image losses and it has drawbacks in converting a 2-dimensional image to 1 dimensional vector while training the model. When ANN gives a prediction it does not give any reason behind it. Sometimes it doesn't give optimum results. ANN can only work with numerical inputs. So, problems need to be translated into numerical inputs.

3.10.2 Convolutional Neural Network

Convolution is a mathematical approach of generating a function associated with two other functions and describing how the relationship between those functions puts an impact on each other's shape and mechanism. The operation of convolution was implemented in sectors of machine learning, specifically in neural networks resulting as a type of neural network specialised in Image Classification and Computer Vision; named Convolutional Neural Network (CNN).

Convolution Neural Networks have been used to analyze visual vision and identify artifacts in photographs, as well as diagnose diverse patterns in time series data and sensor data categorization. A CNN is a feed-forward neural network made up of a series of connected layers, similar to a multilayer perceptron but intended for low computing requirements.

The layers of CNN consist of an input layer, an output layer and a special kind of layer which associates multiple convolutional layers, pooling layers and fully connected layers. The layers are stacked on top of each other in sequence, each one of them is capable of understanding complex shapes, handwritten digits and it is also possible to distinguish individual human faces in specific layers.

The very first layer of the network is the input layer, which corresponds to an N-dimensional array of input data. The special or hidden layers between input and output are basically convolutional layers followed by pooling layers, fully connected (FC) layers and sometimes activation layers. The convolution layer is used to extract various information from the input images and in this layer the mathematical operation of convolution is performed. The layer can be visualized as many small square templates, called Kernel [22] representing the importance given to each pixel of each part the input image is divided in. This kernel applies element wise multiplication between a certain part of the image pixels and a matrix of its own generating a single output.

Convolutional layer is followed by a pooling layer, whose primary concern is minimizing the dimension of data by merging the outputs of neuron clusters to reduce the computational cost. Different types of pulling is used in the entire layer; local pooling for combining small clusters, global pooling for all the neurons in a feature map; there is also max pooling and average pooling, where max pooling uses the maximum value of each local cluster of neuron and average pooling uses the average value [8]. Fully connected (FC) layer is relatively similar to the convolutional layer as it follows the multilayer perceptron too. It is comprised of weights and biases and used to link the neuron between two layers and usually placed before the output layer and completes the integral part of CNN architecture.

Convolutional Neural Networks are widely known for image analysis and pattern recognition but they have also been adapted in a variety of other machine learning applications like Natural Language Processing, Sentiment analysis, Self-driven cars, speech recognition and many more [20].

3.10.3 RNN

Artificial neural networks one of the classes is called Recurrent neural network (RNN). In RNN the nodes have connections between them which makes a directed graph along the temporal sequence and it is used to represent the temporal dynamic behaviour [11]. There are two broad classes of network which are represented by the term recurrent neural network. These classes have similarity in their general structure. One is a finite impulse neural network which is a directed acyclic graph. This can be unrolled and replaced with a strictly feedforward neural network. On the other hand, there is infinite impulse neural network which is a directed cyclic graph and cannot be unrolled. Both finite and infinite impulse neural networks can represent temporal dynamic behaviour.

Infinite impulse and finite impulse neural networks have extra stored states and neural networks can directly control this extra stored state. If any graph or network has feedback loops or timing delays then they can be the replacement of these extra stored states. These controlled states are also known as gated state or gated memory which are a part of recurrent memory and long short-term memory (LSTMs).

Recurrent Neural Networks (RNN) have some varieties some of which are fully recurrent, Elman, Jordan, second order, bidirectional and so on. Fully Recurrent Neural Networks (FRNN) are considered as the most general neural networks topology where inputs of all neurons are connected to the outputs of all neurons. Then there is Elman Neural Network (ENN) which consists of three layers and a set of context units. The context unit is connected to the middle layer and the back connections store the previous value of the hidden unit from the context unit. Moreover, in every time step a new input is given and a learning rule is applied to it.

However, there is another neural network which is Jordan network and here the context unit or state layer is given to the output layer unlike the hidden layer. Another type is Hopfield network which is not a general RNN because it needs stationary inputs and all the connections are evenly sorted. Echo State Network (ESN) is another variant where only the outputs of the neurons can be trained and it has a weakly attached random hidden layer. Then the Independently recurrent neural network (IndRNN) works on the problems of FRNN. Here the neurons in one layer get only their past record as context units and this is how it maintains the independence. This neural network can also be trained with ReLU.

In addition to this there is Recursive neural network which has a differentiable graph type structure. In this structure similar sets of weights are implemented in a loop and the graph-like structure is visited in a topological order. After that there is second order RNN which has states as a product and it uses higher order weights. Moreover, there is Bidirectional RNN which determines each of the elements of the sequence using a finite sequence and considering the past and future context. To do this concatenation is used in this RNN. Lastly, there is Multiple timescale RNN which is a neural based computational model. Through self-organization the functional hierarchy is simulated in this RNN.

3.10.4 LSTM

Long Short-Term Memory (LSTM) refers to an artificial recurrent neural network (RNN) architecture that is a part of a complex area of deep neural networks. This network studies the order dependence in sequence prediction problems.

Recurrent neural network is a subclass of artificial neural networks. In this network the connections of the nodes are directed along a sequence of time. Though it is derived from the feedforward neural networks where the data is being passed from the input to output incapable of flowing backward, recurrent neural networks have feedback loops. Sequences are time dependent series of vector patterns that are used as input and output to the network. [Long Short-term Memory in RNN]. With standard forms of recurrent neural networks, long sequences get complex to master.The reason behind this complexity is because it is conditioned using back-propagation over time (BPTT) which results in vanishing or expanding gradients. RNN generally consists of the units which interact in discrete time through directed graphs with weight [3]. The magnitude of the weights has an exponential effect on the temporal evolution of the back propagated errors. This can happen while the learning to resolve long time delays takes an excessive amount of time or fails completely.

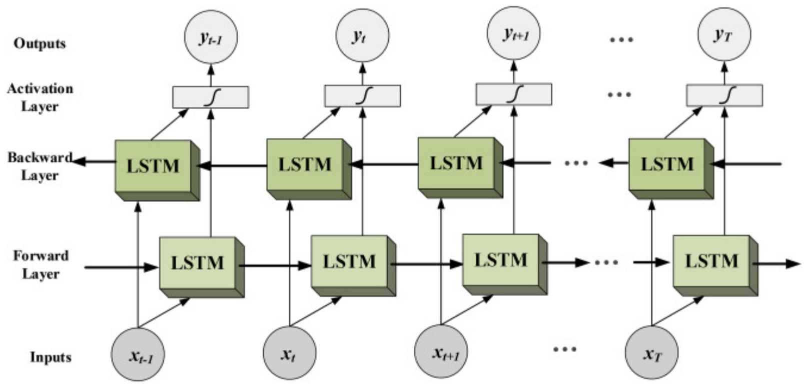

Figure 3.7: Layers of LSTM

The LSTM algorithm, which implements non-decaying error flow backwards in time, is the solution to this problem. This network is capable of resolving long time intervals more than 1000 steps even if the sequences are noisy. LSTM networks are competent for classification, processing, and prediction by the analysis of time series data [3].

A deep LSTM consists of three bidirectional layers - two feedforward layers and a softmax layer. The bidirectional layers are to analyze the temporal dependencies that the RNN network is incapable of [19]. Besides, this network basically consists of memory blocks(cells) known as hidden layers.

There are two states that are transmitted to the next cell are the cell state and the hidden state. The cells store the information. The modifications in the cells are executed by three major gates- input gate and output gate and a forget gate. The cell preserves attributes throughout arbitrary time spans. The three gates control the transfer of data from and to the cell.

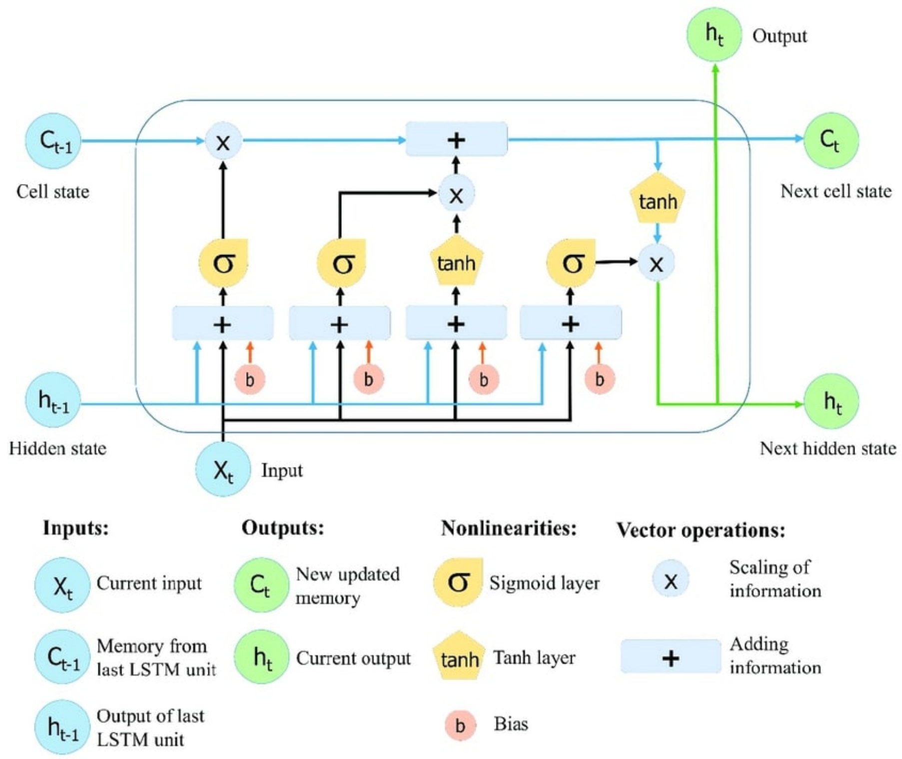

Figure 3.8: Three Gates of LSTM

In other words, the gates differentiates between which information to be remembered and which are not. Besides, a memory is added in the cell so that it can retain the antecedent sequences. In order to prevent long-term dependence problems, forget gates were added in LSTM. The function of the first gate, a forget gate,is to determine which piece of information is to be discarded from the cell state and which one to keep. The sigmoid layer makes this decision. Input gate is the second gate consisting of a sigmoid layer for determining which values are to be modified. Furthermore, the current state output is defined by the modified cell state and the sigmoid layer which holds the information of which cell state components will be the final cell output [3].

Chapter 4

Methodology

BCI technology provides an interface between computer and human brain through a device or machine to communicate with each other for people with significant mobility limitations. Electroencephalography is recognized to be the most realistic and plausible non-invasive BCI method considering the precision, convenience of signal acquisition and cost effectiveness relative to other technologies. The main purpose of our work is to use the dataset collected from the cap to grasp and hold any object and move the limb in the direction the users want. In order to do so, the system is needed to design a process where the EEG machine takes the necessary information from the cap as input data, respectively processing the input data, analyzing and classifying those data and producing the required output like move right, move left, lift up etc.

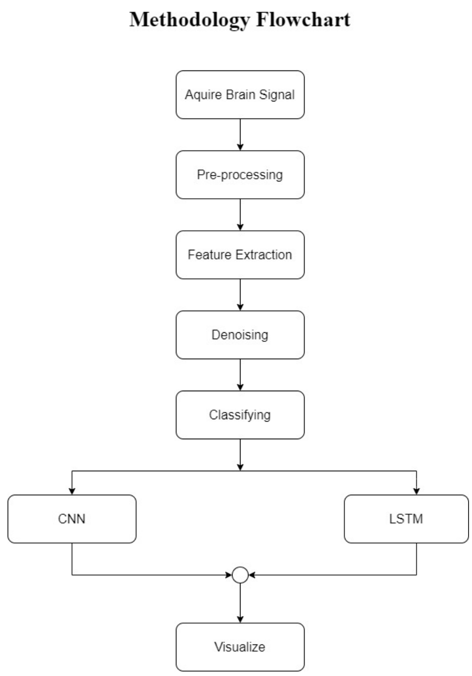

Figure 4.1: Methodology Overview

The whole process is consists of three major phases:

4.1 Signal Acquisition

Signal acquisition technique records brain cortical activity. This process mainly refers to the system where the signals or brain waves are collected through the electrodes attached to the cap. Some use 5-channel bipolar electrodes and others use 12 Ag/Cl electrodes. Simpler commercial BCI with one electrode and an on-ear reference electrode are also used. Signals are obtained by placing the electrode at a specific location in the brain, often using a 10-20 positioning system. The electrode uses a 10-10 scheme, which is a much more dense electrode placement technique.

4.2 Pre-processing

The data collected from the signal processing phase contains a lot of noise. In this phase, the noise is eliminated. Filtration of the signals is the most commonly used pre-processing method. CSP and PCA are often used to reduce the ratio of signal to noise. P300 EEG signals are selected by visually distinguishing between the P300 and non-P300 signals. Until estimating the power spectral density of the signal every 62.5 millisecond, the 64 channel was spatially filtered using a standard mean guide. In the calculated signal, an ideal weight vector is combined linearly to cancel out noise.

4.3 Feature Extraction and Classification

Feature extraction is a dimensionality reduction procedure that compresses a wide range of raw data into small distinct groups for analysis and processing. The process of feature extraction starts with recording the cortical activities of the brain. Electrodes are placed all over the head and earlobe to acquire the signals from the brain and later processed as data. After the pre-processing, the recorded EEG data are classified with a bandpass filter that distinguishes the bands with a range of 1 Hz from a center frequency. To improve the signal to noise ratio, common average reference (CAR) spatial filtering is used on the obtained raw brain signals.

$$e_i(t) = e_i(t) - \frac{1}{N} \sum_{i=1}^{N} e_i(t) \mid$$

Here $e_i(t)$ is the value of the signal for the i channel at time t and N is the number of channels.

One of the most essential steps of our work is pre-processing; which is comprised of artifact removal from the raw signals, signal augmentation and edge identification. The next step of this operation is the feature extraction scheme. This is meant to select the information and signals that are more relevant to incorporate with the algorithm. There are a few established methods for Feature extraction; our paper presents some mathematical models for extracting features from the obtained EEG signals and describing their characteristics.

4.4 Fast Fourier Transform (FFT) Method

The Fast Fourier Transform method mainly decomposes the complex raw signals into smaller divisions and prepares the signals ready to be analyzed. After the analysis, the characteristics of the EEG signals are computed by power spectral density (PSD) estimation, where we choose our data to be processed further. PSD is derived by a Fourier transforming method named Welsh Method. FFT is more useful in terms of real life applications considering it has an optimal time complexity than most other techniques of feature extractions. But it is also quite sensitive while processing signals having a wide bandwidth.

4.5 Wavelet Transform Method

Wavelet transformation is considered to be the most suitable method for EEG data extraction because it distributes the signals into smaller time-frequency domains for better analysis. Wavelet transform represents the signal in a precise way using short time windows and works better with both low frequency and high frequency resolutions. In this method the wavelets are demonstrated as simple build blocks for easier mathematical approach and even better suited for handling any sort of irregularities in the signals, uneven pattern in the impulses, accidental and sudden changes in the wavelets in various time domains. Wavelet Transform method is further classified into two other patterns namely, Continuous Wavelet Transform and Discrete Wavelet transform for better precision and accuracy in the models.

4.6 Eigenvectors

Eigenvectors are usually associated with linear algebra or linear systems of equations. However, it is also considered to be a set of approaches concerned with determining the frequency and amplitude of attained and preprocessed signals from the brain. There are some methods, namely Pisarenko's method, MUSIC method, and minimum-norm method are known as eigenvector methods, as these methods are strongly interrelated with calculating strength and characteristics of signals. Among them Pisarenko's method is used to analyze Power Spectral Density (PSD). Power Spectral Density is the fluctuation in the frequency that occurs inside an oscillatory signal which is measured as frequency per unit of mass. PSD is calculated by defining coefficients of the equation and taking the value of inversion of the squared coefficient. MUSIC method annihilates false zero concerns by using signal's average equivalent to the domain of the entire eigenvector. To derive a specified noise domain vector a from either the noise or signal domain eigenvectors, the Minimum Norm method generates false zeros in the curve to differentiate them from true zeros. The Pisarenko method, on the other hand, utilizes only the noise domain eigenvector subsequent to the minimum eigenvalue, whereas the minimum norm technology requires a linear function of all noise domain eigenvectors [12].

4.7 Autoregressive Model

Using a parametric approach, autoregressive (AR) methods calculate the EEG's power spectrum density (PSD). As a result, unlike nonparametric approaches, AR methods do not suffer from spectral leakage, resulting in improved frequency resolution. The coefficients, or parameters, of the linear system under consideration are calculated for PSD estimation.

Furthermore, during the movement, Root mean square of P300 like signal and Root mean square of signal obtained during different command flashes are compared, and the average profile of which corresponding RMS value is smallest is selected as command. A threshold condition is defined and if the threshold condition is satisfied the system sends a control command, and the second condition is that the same button should both be recognized by both P300 and SSVEP three times and then the command is sent out. In P300 and ERD is combined to control the movement of the limb and the EEG signals are first spatially filtered with common average reference and then the signal is filtered using a band pass filter at 8-32 Hz, spatial pattern is computed from the signal.

4.8 Denoising

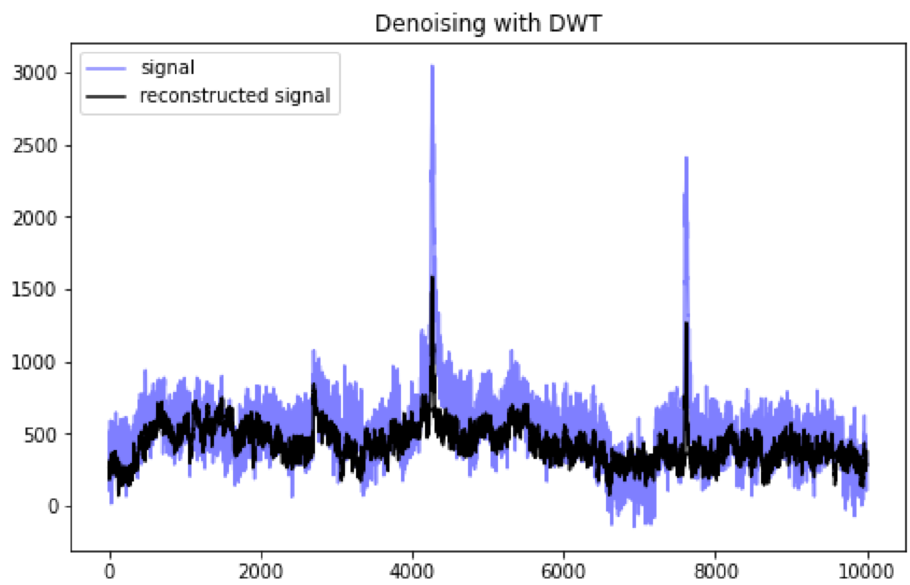

Figure 4.2: Denoising Signals

In signal processing, noise is considered as irrelevant information in a signal. To remove the noise denoising method is used. It is a signal processing method that extracts signals from a mixture of signal and noise so that it can preserve the useful information.

The electrical activity of neurons in the brain produces an electroencephalogram (EEG). Since these signals have such small amplitudes, they are easily altered by various noises. Because of the existence of noises, EEG analysis for evaluation and information is difficult. A variety of methods and techniques to deal with these noises have been developed. Regression, blind source isolation, wavelet and empirical mode decomposition are some of these approaches.

In our code, to decompose the signal, in the wavelet_denoising method an ordered list of coefficient arrays (coeff) is declared. Through the pywt.wavedec( ) function decompose the given signal to 1D multilevel Discrete Wavelet Transform (dwt). In the function the parameters are respectively data (x), wavelet (a name) and mode which is set to per (periodization) so that the smallest possible number of the decomposition coefficients can be received. Then the standard deviation (sigma) is being calculated to measure the total amount of dispersion in the set of values. The calculation is done by taking the mean absolute deviation of a signal. After that the threshold (uthresh) is calculated by multiplying sigma with np.sqrt() (Numpy sqrt()) function. Here in the np.sqrt() function the parameters are np.log() which is a mathematical function used to calculate the logarithm of x and this is multiplied by 2. Now by coeff[1:] we have reconstructed the signal but leaving out the high frequency part of the signal. This filtering out the high frequency is done by the pywt.threshold() function where the parameters are respectively i, the value is the previously calculated threshold and mode which is set to hard (hard thresholding). Next we have returned the pywt.wavedec() function where the parameters are respectively decomposition coefficient (coeff), wavelet and the mode is per which refers to periodization.

Now to get the mean absolute deviation, there is a method named madev where the parameters are d and axis which is set to none. Afterwards, we returned the np.mean() function which is used to compute the arithmetic mean value. The parameters of this function are np.absolute() function which calculates the absolute value of the given input and another parameter is axis.

4.9 CNN Model Explanation

In the code of CNN at the very beginning we have taken load, time_steps and subsample as 1, 1000 and 50 respectively. Then the model is created in a sequential way where it is a linear stack of layers. The sequential API gives the flexibility to create models layer by layer but can not share layers. Now we added the first layer which is the 2D convolution layer using model.add(Conv2D)) method where the first parameter is filters=120 which is the receptive field and means that it is the number of filters the convolutional layer will learn. Then there is kernel_size= (7,7) which refers to the width and height of the 2D convolution window. The parameter padding= “same” is used to keep the spatial dimension of the volume so that the output and input volume size are the same. Then the parameter activation= “relu” refers to the activation function which we want to apply after the convolution is done. Here the term “relu” means rectified linear activation function. This function will directly output the input if it is positive, or else if the output is not positive it will provide 0 as the output. In the first layer we need to mention the input shape which says the shape of one sample. After that, we have added the batch normalization layer by model.add(BatchNormalization()), where it is used to standardize the input or outputs. Then we have added max pooling to our convolutional layer which is used to reduce the resolution of the output that we have got from the layer. In the max pooling operation pool size defines the size of the input window for each channel of input. In our code pool_size(3,3) means it will take the maximum value of a 3x3 pooling window.

Afterwards we added the second 2D convolutional layer where the parameters are filters=120, kernel_size= (5,5), padding= “same” and activation= “relu”. Here the layer will learn 120 filters and the width and height of the convulational window is set by the kernel size. Moreover, the padding is set as “same” to keep the output and input volume size similar and the activation is set as relu (rectified linear activation function) as previous, which will give the input as the output if it is positive, otherwise it will provide 0 as output. Then the batch normalization is added for the second layer which standardizes the input or output.

Then we have again added the max pooling to our convolutional layer to summarize the output of the convolution layer. Following that we have added the third 2D convolutional layer where the parameters are filters=64, now the kernel_size is (3,3), padding remains “same” and activation is “relu”. Here the layer will learn 64 filters and the width and height of the convulational window is set by the kernel size. Moreover, the padding is set as “same” to keep the output and input volume size similar and the activation is set as relu (rectified linear activation function) as previous, which will give the input as the output if it is positive, otherwise it will provide 0 as output. Then the batch normalization is added for the third layer which systemizes the input or output. After completing the previous steps the next thing we are doing is flattening our pooling layers by adding a flatten layer which is model.add(Flatten()). In other words with the flatten function we are flattening the inputs to a single column or we can say the multidimensional tensor(arrays) gets serialized and converts it into a single array after this layer is applied.After that, this long vector of input data is passed through the neural network for further processing. Furthermore, the dense layer uses this feature vector for the final classification. The next step is to add a dropout layer which is defined by model.add (Dropout (0.2)). We add this layer to overcome the overfitting, an issue which usually happens during the training of the model.In this case the model becomes dependent on the dataset while making a prediction and does not give satisfactory outcomes. The dropout layer works by decreasing the odds of overfitting by dropping some neurons randomly in each stage. The 0.2 defines the drop rate of the neurons by this layer which is here 20%. Afterwards we have added a dense layer by model.add(Dense(32, activation = ”relu”)). Here the dense layer refers to the fully connected layer where all the neurons of a layer are connected to the following layer. Besides, 32 specifies the number of the neurons present in the current layer.Furthermore, for the activation function relu (rectified linear unit) has been used.

For the next part of our implementation, we have incorporated an optimization process that implements the adam algorithm. Here adam computes individual learning rates for different parameters and passes the default value of learning rate, lr=0.0001 through the adam optimizer method. Then we have added a compile method which configures the learning method before the model is trained which is basically the compilation process before the code is ready to process. The compile method is receiving three arguments: an adam optimizer, a loss function categorical_crossentropy and a list of metrics, here used accuracy and mse. Lastly, we have used a summary method to summarize and sum up the whole process of creating a CNN model, adding layers and compilation.

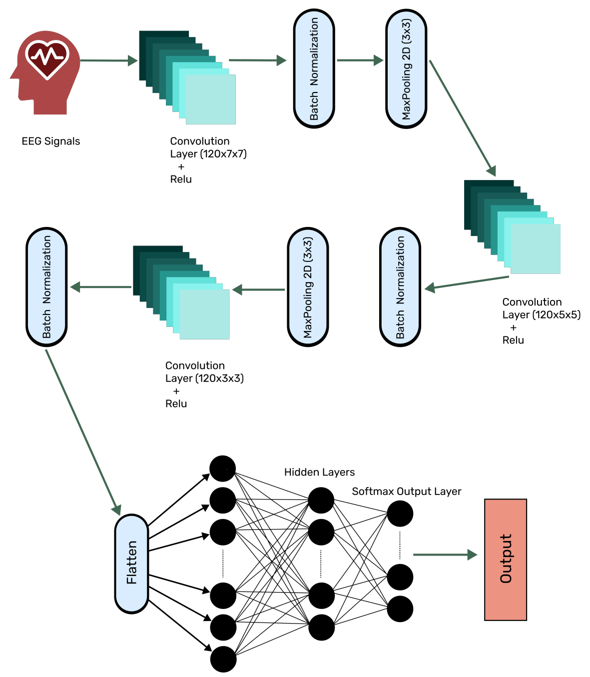

Figure 4.3: Layers of CNN

4.10 LSTM Model Explanation

At the beginning of the code of LSTM, we have taken time_steps and subsample as 1000 and 10 respectively. After that the model has been defined as a sequential model stacking multiple layers. To add the first layer, firstly the number of units has been defined as 64. The number of units refers to the number of the neurons that are linked to the layer where hidden state and input get concatenated.It also denotes the dimension of the cell's outer space.

The next parameter that is passed is input_shape argument which specifies the first hidden layer of the network. In this network, the input of every layer must be three dimensional,therefore,to reshape the input data a tuple holding values of steps and number of input units is passed. The values in this model are floor values of time_step/subsample and 32 respectively.

To connect the three-dimensional input, output of the previous hidden state, to the next layer of LSTM return_sequences is set to True. After that a dropout layer is added in the code section. A dropout layer prevents the overfitting of the model and robust the model's performance. The layer randomly drops some connections between two layers. Furthermore, the dropout rate is specified while adding the layer. In this model the rate is 0.5 which means that 50% of inputs is set to zero.

Additionally, another layer is added to the network. In this layer, the number of unit is 32. Furthermore, to reshape the input data, input_shape argument is specified which is the same as the previously added layer. The return sequence is set to True so that the hidden state output returns the sequence for every input and the following layer can receive the sequence as input. To improve the performance again a dropout layer is added with a rate of 0.5.

After that, a layer with 16 units is added. The following line of the code represents a fully connected layer added to the model. The first parameter of this layer represents the number of neurons in each layer which is 6 for this model. The other parameter is activation. The neurons of the human brain get activated after a certain potential and this threshold is called activation. Here sigmoid function is used as the activation potential.

Moreover, it is necessary to set up the learning process before training the model. To do that a compile method is called containing three parameters. The first parameter is loss function. This function investigates how well the algorithm is modeling the given dataset. As we have used sigmoid as an activation function, the most compatible loss function is Binary Crossentropy. The second parameter is an optimizer which is used to reduce the losses of the model. Here Adam optimization algorithm is used and the learning rate is 0.0001. The third parameter is metrics. In this model accuracy metric is used. This metric estimates the accuracy rate of all possible predictions. Lastly to summarize the whole model, model.summary() is called.

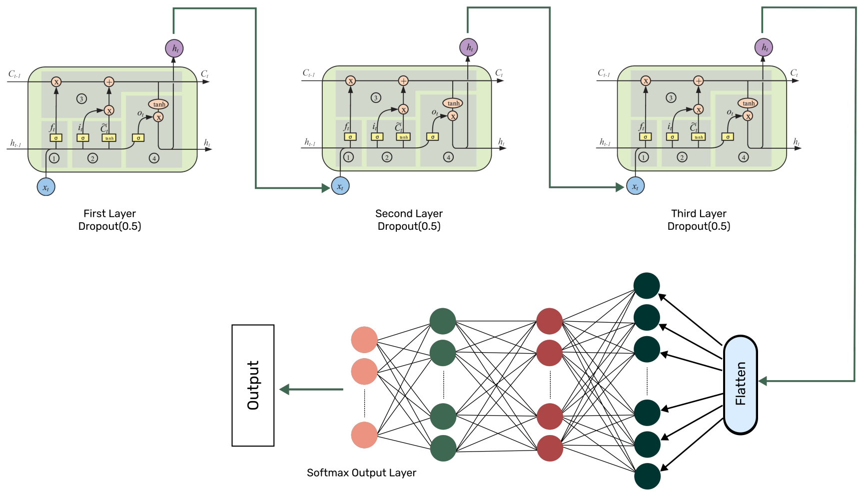

Figure 4.4: LSTM Model

4.11 KNN Model Explanation

To generate the model, we have to build the classifier of the model. At first, we have to import the module of Kneighbors Classifier. After creating the module, we have created a KNN classifier object named neigh by using Kneighbors Classifier (n neighbors = 3). The argument that is passed in the function represents the number of neighbors this function will create. Then we have to train the model by using our training sets. To do so, we will use a fitness function with the help of the object we have created. After training the model now we have to anticipate the response of the test dataset. Therefore, we have used predict function to perform the prediction. In this function the argument is xval. After that, we have written the code for plotting the ROC curve. Now to know the accuracy of our model, we can compare the previous test set values to the predicted values. For that, first, we have to import the metrics module, accuracy score, to calculate the accuracy. Then in the module, we have to pass the values as parameters. Therefore, to show how often our model is correct we have used the accuracy module: accuracy score (yval, predictions).

Chapter 5

Result and Discussion

Google Colab has been used to run all the models. The research commenced with assembling brainwaves for training the models which include brainwaves of adults as well as children. The outcome of the models - CNN, LSTM and KNN in terms of training and validation accuracy and losses are visualized in the results section. In the training accuracy graph the 'Y' axis represents the accuracy percentage and 'X' axis represents the epoch numbers. Blue line is showing the values of training accuracy and the red dotted line is showing the values of validation accuracy.

Similarly, in the graph of training loss, the 'Y' axis is representing the accuracy percentage and 'X' axis is representing the epoch number. Blue line is showing the training loss values and the black line is representing the validation accuracy values. For every 20 epochs the accuracy and loss value is changing. Gradually, the training accuracy percentage has increased and the training loss has come to a consistent level reaching 92%. At first when the training of the model is completed, the validation set is given to the model to bring the necessary modifications to the network. A model will be considered as a better model if validation accuracy is equal or approximate to the value of the training accuracy. Furthermore, when training loss gets increased, it refers to the decrease in validation accuracy.

At the very beginning of the training, In the CNN model, the training accuracy was less than the validation accuracy. It indicates that the model is underfitted as the model is already acquainted with training data, while validation data is a collection of new data sets that are unfamiliar to the model. To improve the model we have added more layers and neurons. After an amount of training the training accuracy is greater than the validation accuracy.

At one point the graph depicts that the difference between the training accuracy and the validation accuracy is very less. At some points, the training accuracy and the validation accuracy are even equal. Therefore, as the epoches increased, training accuracy was also increasing and at the end of the training the model accuracy of CNN was 90.30% and the testing score that we found was 90.11% . In addition, the computation time of the model was 690ms per step.

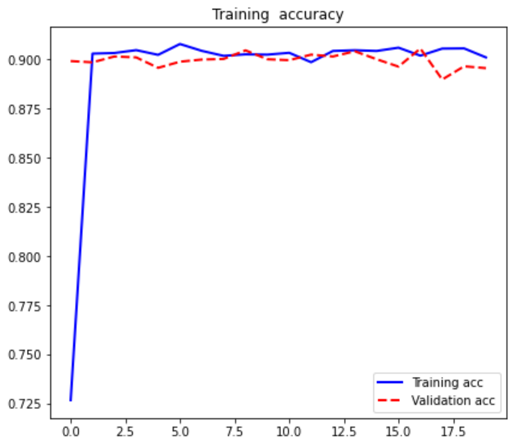

Figure 5.1: CNN Accuracy Graph

For the LSTM model, the same procedure has been followed. In our LSTM model, we have the same parameters in the horizontal and vertical axis as CNN. The graph shows that the difference between the training accuracy and the validation accuracy is slightly greater than the CNN. However, the difference is still less and in some points, it tends to be zero. Between 30 to 40 epochs we have got the highest accuracy of our model.

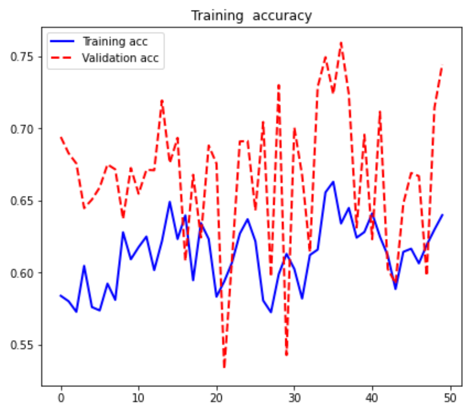

Figure 5.2: LSTM Accuracy Graph

In between these two epoch times, the training accuracy was around 64.91% where the validation accuracy got higher than the training accuracy. The validation accuracy was also highest at these epochs which is 74.44%. The performance of this model is marginally less than the CNN. The time required for this model to perform the computational process is 267ms per step. Moreover, the accuracy of this model is less than CNN as well as KNN. In comparison to the accuracy of the three models, CNN has performed better than both LSTM and KNN.

Models	Training Score	Testing Score
CNN	90.30%	90.11%
LSTM	64.91%	74.44%

Table 5.1: Accuracy Comparison of Neural Models

Moreover, there are some more models named as Random Forest, Adaboost Classifier, XGB Classifier, KNN, MLP. For the random forest model the testing accuracy is 96.02% and the time it is required is 1.061s. Then adaboost classifier has 95.63% accuracy with a testing time of 0.365s, XGB classifier has 96.04% accuracy with 0.061s testing time. After that KNN model has 95.04% testing accuracy and the time it needs is 111.16s and lastly MLP needs 0.36s for testing and provides accuracy of 93.8%. Among all these models XGB classifier provides the highest accuracy which is 96.04% within a lesser time of 0.061s.

Models	Testing Accuracy	Testing Time in Seconds
Random Forest	96.02%	1.061s
AdaBoost Classifier	95.63%	0.365s
XGB Classifier	96.04%	0.061s
KNN	95.04%	111.16s
MLP	93.8%	0.36s

Table 5.2: Comparison between Classifiers

So if we plot these accuracies in a graph we can see a visual representation of the difference of the accuracies depending on different models.

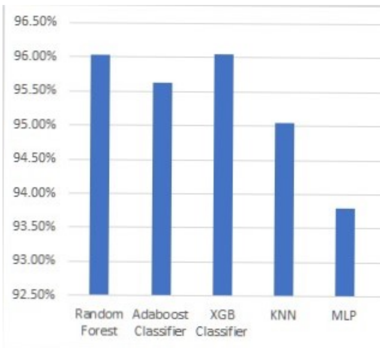

Figure 5.3: Bar chart of Comparison between Classifiers

Chapter 6

Conclusion

There are numerous other organizations and researchers who are working on developing brain-controlled grasp and lift technology and a good percentage of them have successfully implemented different user-friendly features in the whole system. Datasets of the brain waves are collected through an EEG device and the relation between different states of the brain are analysed for finding a threshold value to operate the prostheses. The microcontroller and motor driver in the robotic arm plays a vital role in navigating the arm according to the brain signal command of the user. Though the grasp and lift technology comes with a bunch of features to ease the movement of the user, there are still some limitations that are yet to be resolved. To improve the model of the existing system, we have proposed to add the features of strong grip control, smooth lift and acceleration control. Most of the time it takes a while to start taking action after the brain signals have been generated; that's why if a user suddenly spots an object and decides to catch and hold, there is a narrow possibility of sudden and smooth operation due to execution delay. We are proposing a system that will help the user to take immediate action and will lessen the execution delay effectively. In addition, a programmed microcontroller will be attached to the system to monitor and control the strength of grip. In addition, this initiative also offers a new and successful approach for the improvement of movement for physically impaired people suffering from amputations due to accidents, dysmorphic arm or neuromuscular disease.

The primary goal of our research paper is to develop a better understanding of the models that have been used to incorporate grasp and lift detection. The comparison between all the implemented models gives a general idea of the accuracy of each model and that leads us to find the precise result for the system. Our future scope of this research involves the successful detection of grasp and lift with higher accuracy rate in greater degree paralysis. We would also like to incorporate an even larger dataset to minimize the noise to signal ratio as the brain signals we have acquired don't only contain the signals for grasp and lift only. So the filtration process of the brain signals would be more emphasised. We are heading towards the goal to assimilate the grasp and lift in exoskeleton, eg. robotic arms to assist individuals with amputations or upper limb paralysis cases. On a much broader aspect, we could also indulge our system in the next generation Mars rover, Mongol Tori, so that the robotic arm attached to it can operate a better grasp and lift with the assistance of the preprocessed human brain signals rather than battery controlled devices to operate it. As we are simulating our system solely with the help of brain signals, it would save the user from the hassle of devices and wires clinging to their body which is also energy efficient and convenient in terms of expenses.

Bibliography

[1] A. Bain, *Mind and body: The theories of their relation*. D. Appleton, 1873, vol. 4.

[2] F. Rosenblatt, "Principles of neurodynamics. perceptrons and the theory of brain mechanisms," Cornell Aeronautical Lab Inc Buffalo NY, Tech. Rep., 1961.

[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. DOI: 10.1162/neco.1997.9.8.1735.

[4] J. R. Wolpaw, N. Birbaumer, W. J. Heetderks, D. J. McFarland, P. H. Peckham, G. Schalk, E. Donchin, L. A. Quatrano, C. J. Robinson, T. M. Vaughan, *et al.*, "Brain-computer interface technology: A review of the first international meeting," *IEEE transactions on rehabilitation engineering*, vol. 8, no. 2, pp. 164–173, 2000.

[5] A. Subasi and E. Ercelebi, "Classification of eeg signals using neural network and logistic regression," *Computer methods and programs in biomedicine*, vol. 78, no. 2, pp. 87–99, 2005.

[6] A. C. N. Society *et al.*, "A proposal for standard montages to be used in clinical eeg," *Obtenido de http://www. acns. org/pdf/guidelines/Guideline-6. pdf*, 2006.

[7] N. Glossas and N. Aspragathos, "A cluster based fuzzy controller for grasp and lift fragile objects," in *18th Mediterranean Conference on Control and Automation, MED'10*, 2010, pp. 1139–1144. DOI: 10.1109/MED.2010.5547640.

[8] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Sequential deep learning for human action recognition," in *International workshop on human behavior understanding*, Springer, 2011, pp. 29–39.

[9] J. S. Kumar and P. Bhuvaneswari, "Analysis of electroencephalography (eeg) signals and its categorization–a study," *Procedia engineering*, vol. 38, pp. 2525–2536, 2012.

[10] V. Lawhern, W. D. Hairston, K. McDowell, M. Westerfield, and K. Robbins, "Detection and classification of subject-generated artifacts in eeg signals using autoregressive models," *Journal of neuroscience methods*, vol. 208, no. 2, pp. 181–189, 2012.

[11] K.-L. Du and M. Swamy, "Recurrent neural networks," in. Dec. 2014, pp. 337–353, ISBN: 978-1-4471-5570-6. DOI: 10.1007/978-1-4471-5571-3_11.

[12] A. S. Al-Fahoum and A. A. Al-Fraihat, “Methods of eeg signal features extraction using linear analysis in frequency and time-frequency domains,” *International Scholarly Research Notices*, vol. 2014, 2014.

[13] A. B. Hassanat, M. A. Abbadi, G. A. Altarawneh, and A. A. Alhasanat, *Solving the problem of the k parameter in the knn classifier using an ensemble learning approach*, 2014. arXiv: 1409.0919 [cs.LG].

[14] N. Wang and M. R. Lyu, “Extracting and selecting distinctive eeg features for efficient epileptic seizure prediction,” *IEEE journal of biomedical and health informatics*, vol. 19, no. 5, pp. 1648–1659, 2014.

[15] J. An and S. Cho, “Hand motion identification of grasp-and-lift task from electroencephalography recordings using recurrent neural networks,” in *2016 International Conference on Big Data and Smart Computing (BigComp)*, 2016, pp. 427–429. DOI: 10.1109/BIGCOMP.2016.7425963.

[16] G. Biau and E. Scornet, “A random forest guided tour,” *Test*, vol. 25, no. 2, pp. 197–227, 2016.

[17] A. Hamad, E. H. Houssein, A. E. Hassanien, and A. A. Fahmy, “Feature extraction of epilepsy eeg using discrete wavelet transform,” in *2016 12th International Computer Engineering Conference (ICENCO)*, 2016, pp. 190–195. DOI: 10.1109/ICENCO.2016.7856467.

[18] S. K. Swee, K. D. T. Kiang, and L. Z. You, “Eeg controlled wheelchair,” in *MATEC Web of Conferences*, EDP Sciences, vol. 51, 2016, p. 02 011.

[19] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, and X. Xie, “Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, 2016.

[20] I. Namatēvs, “Deep convolutional neural networks: Structure, feature extraction and training.,” *Information Technology & Management Science (Sciendo)*, vol. 20, no. 1, 2017.

[21] T.-T. Wong and N.-Y. Yang, “Dependency analysis of accuracy estimates in k-fold cross validation,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 11, pp. 2417–2427, 2017.

[22] C. Garriga Estradé, “Studying the characterization of deep cnn neurons,” M.S. thesis, Universitat Politècnica de Catalunya, 2019.

[23] M. A. Gull, S. Bai, and T. Bak, “A review on design of upper limb exoskeletons,” *Robotics*, vol. 9, no. 1, p. 16, 2020.

[24] A. Nair, *A beginner’s guide to scikit-learn’s mlpclassifier*, Nov. 2020. [Online]. Available: https://analyticsindiamag.com/a-beginners-guide-to-scikit-learns-mlpclassifier/.

[25] F. Rustam, A. A. Reshi, I. Ashraf, A. Mehmood, S. Ullah, D. M. Khan, and G. S. Choi, “Sensor-based human activity recognition using deep stacked multilayered perceptron model,” *IEEE Access*, vol. 8, pp. 218 898–218 910, 2020. DOI: 10.1109/ACCESS.2020.3041822.

[26] X.-D. Zhang, “Machine learning,” in *A Matrix Algebra Approach to Artificial Intelligence*, Springer, 2020, pp. 223–440.

[27] S. Ali, "Brain-computer-interfacing & respondeat superior: Algorithmic decisions, manipulation, and accountability in armed conflict," *Catholic University Journal of Law and Technology*, vol. 29, no. 2, pp. 1–30, 2021.