Republic of Iraq
Ministry of Higher Education
And Scientific Research
University of Babylon
College of Engineering
Electrical Engineering Department

Brain Signal Classification in Stroke Using Machine Learning

(Electroencephalography and Magnetic Resonance Imaging)

A Dissertation

Submitted to the Electrical Engineering Department/ College of Engineering/ University of Babylon In Partial Fulfilment of The Requirements For The Degree of Doctors of Philosophy In Electrical Engineering /Electronics & Communications

By

Riyadh Abdulhamzah Mohammed Al-Alwani

Supervised by

Prof. Dr.

Kasim Karam Abdalla

Prof. Dr.

Farah Nabil Abbas

2024A.D **1445 A.H**

Supervisor Certification

We certify that this dissertation entitled (**Brain Signal Classification in Stroke Using Machine Learning (Electroencephalography and Magnetic Resonance Imaging)** and submitted by a student (**Riyadh Abdulhamzah Mohammed Al-Alwani**) was prepared under my supervision at the Department of Electrical Engineering, College of Engineering, University of Babylon, as a part of the requirements for the degree of Doctor of Philosophy in Electronics and Communications Engineering.

Signature:
Supervisor*: *Prof. Dr. Kasim Karam Abdalla
Date: /2/ 2024

Signature:
Supervisor: ***prof. Dr. Farah Nabil Abbas***
Data: / 2/2024

I certify that this dissertation mentioned above has been completed in Electronics and Communications Engineering in the College of Engineering/University of Babylon.

Signature:
Head of Department: Prof. Dr Qais Kareem Omran
Date: / / 2024

Examining Committee Certificate

We certify that we have read this dissertation entitled "**Brain Signal Classification in Stroke Using Machine Learning (Electroencephalography and Magnetic Resonance Imaging) "** and as an examining committee, we examined the student "**Riyadh Abdulhamzah Mohammed Al-Alwani**", in its contents and that in our opinion it meets the standard of a dissertation for the degree of **Doctor of Philosophy** in Electrical Engineering/ Electronics and Communications Engineering.

Signature: **Name:*prof.Dr samir jasim mohammed*** **(Chairman)** **Date: / /2024**	**Signature:** **Name:*Prof.Dr. Bayan Mahdi Sabbar*** **(Member)** **Date: / /2024**
Signature: **Name:*Prof.Dr.Ibrahim A Murdas*** **(Member)** **Date: / /2024**	**Signature:** **Name:*Prof. Dr.Saad Saffah Hassoon*** **(Member)** **Date: / /2024**
Signature: **Name: *Prof. Dr Muhannad Yahya Idrees*** **(Member)** **Date: / /2024**	

Approval of Head of Department	**Approval of the Dean of Collage**
Signature: **Name: *Prof. Dr Qais Kareem Omran*** **Date: / /2024**	**Signature:** **Name: *Prof. Dr Laith Ali Abdulraheem*** **Date: / /2024**

Abstract

Worldwide, cerebrovascular accidents, or strokes, account for the vast majority of deaths. Therefore, a major focus of current study is the identification of symptoms that may occur before a stroke. This research aims to improve the effectiveness of life-saving therapies for high-risk individuals by addressing the cited problem. The goal is to help people feel better and more hopeful again.

Stroke is one of those things that many people experience and doesn't really prepare you for. Therefore, people don't have a lot of time to do anything before it occurs, including preparing for it. Stroke survivors may experience profound changes in their physical skills, perceptions, and cognitive capacities. Many different things may happen to people after a stroke, but certain consequences seem to be more common than others. This brochure was created with the intention of informing individuals about strokes and how they might impact their daily lives.

Attacks on the brain are medically known as strokes. A lack of oxygen reaches brain tissue, causing this condition. important A person may have both ischemic and hemorrhagic strokes, which are subtypes of the same disease. The first sort of stroke to occur when blood supply is momentarily cut off to the brain is known as a transient ischemic attack (TIA). Many people use the phrase "mini-stroke" to characterize this condition. In contrast to the more hazardous hemorrhagic stroke, this kind of stroke is often curable.

Electroencephalography (EEG) is a time series technique that is used as an example in this study, while magnetic resonance imaging (MRI) is an example of an image processing route. There are pros and cons to both approaches; but, because to the widespread availability of EEG equipment and chips, the former

(EEG) may be used in any nation, whilst the later (MRI) cannot. Greater in size, complexity, and maintenance requirements.

Using cutting-edge tools like Python 3 and several algorithms linked to the block diagram suggested in Chapter 3, this work employs a two-pronged strategy for detecting strokes through EEG and image processing. The procedure's flow is subsequently described in detail.

The random forest algorithm is the bast and gets more accuracy result (97%) and it's a high value, in the IP way the suggested procedure that explained in chapter three and remember in chapter four gets a good result that explains in table 4.5 and gets 98.1% out of 1000 as a correct results.

The dataset was collected from two distinct sources: the al-Hilal and AL imam Alsadiq hospitals, which were specifically used with the IP path, for MRI images; and the Google location mentioned in Chapter 3 with more explanation, which was used with the EEG route. These two hospitals made contributions to the dataset's compilation.

Remember that both methods employed Python, machine learning, and specialized algorithms. These methods have to provide stroke detection. As long as they worked, additional algorithms and methods might be added to the development process.

Contents

Abstract I

Contents III

List of Figures VII

List of Tables X

List of Abbreviations XI

The list of symbols XIV

Chapter One 1

Introduction 1

1.1 Preface 1

1.2 several key algorithms in this work 3

1.3 Stroke in the Human 5

1.3.1 Transient ischemic attack (TIA) definition 6

1.4 Literature survey 6

1.4.1 Summary of the Survey 13

1.5 Statement of the Problem 17

1.6 The Aims and Requirements of the Work 18

1.7 Contribution in this study 18

1.8 The System's Proposed Restrictions 19

1.9 Formulating Dissertation Strategies 20

Chapter Two 22

Background of Signal and Image Processing for detecting strokes 22

2.1 Introduction 22

2.2 Strokes on the body 22

2.3 Electroencephalography path 23

2.3.1 Advantages of Electroencephalography (EEG) 26

2.3.2 Disadvantages of Electroencephalography (EEG) 26

2.3.3 The representation EEG channel montage 27

2.3.4 Artifacts 28

2.4 Artificial intelligence overview 29

2.4.1 short introduction to artificial intelligence 29

2.4.2 Data mining 31

2.4.3 Unsupervised learning 31

2.4.4 Semi-supervised learning 31

2.4.5 Reinforcement learning 32

2.4.6 Self-learning 32

2.4.7 Feature learning 32

2.4.8 Models 33

2.4.9 Artificial neural networks 33

2.4.10 Decision trees 34

2.4.11 Support-vector machines 35

2.4.12 Machine learning's bias 35

2.4.13 Machine Learning Definition 35

2.4.14 The Mechanisms of Machine Learning 35

2.4.15 The Importance of Machine Learning 36

2.4.16 Machine Learning Is Widely Adopted 36

2.4.17 Introduction to Data in Machine Learning 37

2.4.18 Machine Learning data splitting 38

2.4.19 Properties of Data 38

2.4.20 Understanding Data Processing 39

2.5 Image processing is defined 40

2.5.1 Image Processing Basics 40

2.5.2 Digital Image Fundamentals (DIF) 41

2.5.3 Applications of Digital Image Processing 42

2.5.4 Essential Procedures for Processing Digital Images 42

2.5.5 Representation and Description 44

2.5.6 Sampling and Quantization 44

2.5.7 Thresholding 45

2.5.8 Image Segmentation Computer Science 46
2.5.9 Classes of Segmentation Techniques 46
2.5.10 Convolution Neural Network (CNN) 46
2.5.11 Foundation of Convolutional Neural Network 47
2.5.12 Concepts of Convolutional Neural Network 48
2.5.13 Activation Functions 49
2.5.14 Normalization. 50
3.1 Introduction 52
3.2 General Work Headline 53
3.3 Several factors point in that direction 55
3.4 The dataset description 57
3.5 The stroke-related EEG backdrop 58
3.6 Strategy and Method 60
3.7 Data Pre-Processing 63
3.7.1 Data referencing details 63
3.7.2 Grouping by Type 64
3.7.3 Artifact Rejection 65
3.8 Feature Extraction 67
3.8.1 Time Domain Features 67
3.8.2 Frequency domain features 68
3.9 Classifiers Algorithm Types 73
3.9.1 A Decision Tree 73
3.9.2 Extra-Tree Classifier 73
3.9.3 MLP is a Multi-Layer Perceptron. 74
3.9.4 Random Forest (RF) 75
3.10 Feature Extraction Process Flow Diagram 76
3.11 Second technique for image processing 80
Chapter four 87
Results and Discussion 87

4.1 Introduction 87

4.2 The Electroencephalography (EEG) Based results. 87

4.2.1 Data Description with Attributes 88

4.3 Algorithms That Used in this work 96

4.4 some important concepts 98

4.4.1 Receiver Operation Curve (ROC) 98

4.4.2 The Confusion Matrix (CM) 99

4.5 The Strock Detection Results from EEG path 99

4.5.1 Accuracy: 102

4.5.2 The ROC Analysis with Results 103

4.5.3 Confusion matrix analysis with results 105

4.5.4 classification results 108

4.6 The Image Processing (IP) results 108

4.6.1 Image Acquisition 109

4.6.2 Image Pr-processing 110

4.6.3 Segmentation 110

4.6.4 Binary Image conversion and feature extraction 111

4.6.5 Logistic Regression Classifier 112

4.7 Result Analysis 115

4.7.1 Dataset and System Configuration 115

4.7.2Analyses Based on Experiments 115

4.8 some conclusion in that path 117

4.9 Image Processing Implementation with Results 118

4.10 Compare Between Two Paths 122

Chapter Five 124

Conclusion and Future Work suggestion 124

5.1 Introduction 124

5.3 Future suggestion with Stroke Disease 126

5.3.1 Progress on the EEG Protocol for Stroke Patients in the Future 126

5.3.2 Future suggestion in Image Processing (IP)With Stroke Disease 127
References 128
Appendix-A 1
Appendix-B 1
Appendix (C) 1
Contains 4 Participants With 26 Attributes for Each 1
Appendix-C 1
Appendix-D 1
Appendix-E 1
Appendix-F 1

List of Figures

Figure 1.1 The Block Diagram of The Dissertation 2

Figure 2.1 (a) Hemorrhage stroke (b) ischemic stroke 23
Figure 2.2 An Example of EEG Signals 25
Figure 2.3 standardization location of electrode placement-use standard 25
Figure 2.4 Main types of artifacts in humane 28
Figure 2.5 Machine learning as subfield 30
Figure 2.6 Part of machine learning as subfield of AI or part of AI as a subfield of Machine learning 30
Figure 2.7 An artificial neural network[41] 33
Figure 2.8 simple flow to session tree algorithm[46] 34
Figure 2.9 data processing steps[48] 37
Figure 2.10 data splitting into 3 categories[49] 38
Figure 2.11 data processing steps[51] 39
Figure 2.12 Fundamentals of Digital Image Processing System 41
Figure 2.13 fundamental steps in digital image processing 43
Figure 2.14 image enhancement 44
Figure 2.15 image restoration process 44

Figure 2.16 Schematic diagram illustrating the interconnections between layers in the recognition, Kunihiko Fukushima [66] 48
Figure 2.17 Conceptual model of CNN [67].......... 49
Figure 2.18 Sigmoid function [67] 49
Figure 2.20 Normalization of data[67] 51

Figure 3.1 the main idea behind the suggested way to do the work. 54
Figure 3.2 EEG Channel Frequency bands range 58
Figure 3.3 First Stage Data Cleaning and Feature Extractions 61
Figure 3.4 Second Stage training and testing of machine learning models 61
Figure 3.5 machine learning part procedure 62
Figure 3.6 Blue arrows indicate the frontal and parietal lobe EEG channels C3, C4, O1, and O2. 64
Figure 3.7 the output spectrum to the welch method to the welch.py program . 70
Figure 3.8 display two feature attributes (a) refer to delta band frequency from 1-4Hz (b)Refer to welch spectrum diagram power density 72
Figure 3.9 The Feature Extraction Block Diagram 77
Figure 3.10 (b) flowchart operation of the program 79
Figure 3.11 depicts the phases of image processing in that work. 80
Figure 3.12 Brain stroke in the gray scale image 81
Figure 3.13 Examples of simple structuring elements 83
Figure 3.14 transition of CT scan image to Detect Stroke 84

Figure 4.1 Block diagram of the operation of processing with EEG 87
Figure 4.2 plot all attribute for one participant. 96
Figure 4.3 The algorithms performance accuracy 102
Figure 4.4 Mylti-Layerd Perceptinon Roc Curve. 103
Figure 4.5 Decision Tree Roc Curve 104
Figure 4.6 Extra-Tree Roc Curve. 104
Figure 4.7 Random Forest Roc curve. 105

Figure 4.8 Average Precision Recall and F1-Score of all classifiers. The black line marks 90% benchmark. 108
Figure 4.9 illustrates the proposed model with IP path 109
Figure 4.10 segmentation on the MRI brain using HSV threshold technique .. 111
Figure 4.11 Logistic and Linear regression model 113
Figure 4.12 Linear regression model 114
Figure 4.13 logistc regression ouput program(log.py) 115
Figure 4.14 show the results among all algorithms 116
Figure 4.15 the flowchart operation in the image processing path *1,2,3,4 refer to the subroutine to each task* 119
Figure 4.16 show the image processing steps 120
Figure 4.17 show the original image with morphology for three images 121

List of Tables

Table1.1 Illustrated the Summary of The Literature Survey 14

Table 3.1 dataset details 58
Table 3.2 features summarized 72

Table 4.1 (A, B) Contains 26 Attributes for Four Participants........................... 89
Table 4.2 confusion matrix concept.. 99
Table 4.3 Confusion matrix of all four models' performance withTraining datasets 106
Table 4.4 Confusion matrix of all four models' performance with the test dataset 107
Table 4.5 Real time result comparison.. 117
Table 4.6 Compare Between EEG Path and IP path.. 123

List of Abbreviations

Abbreviated name	**description**
AIS	Artificial Immune System
AL	Artificial intelligent
ANN	Artificial Neural Network
ARM	Auto Regressive Method
AUC	Area Under the Curve
CAA	crossbar adaptive array
CBF	Cerebral Blood Flow
cEEG	continuous electroencephalogram
CNN	Convolution Neural Network
CT	Computerized Tomography
CVA	Cerebrovascular accident
DIF	Digital Image Fundamentals
DL	Deep learning
DOT	diffusion optical tomography
DSS	decision support systems (DSS)
DT	decision tree
DTI	Diffusion tensor imaging
ET	Extra Tree
ECML	European Conference on Machine Learning
ECoG	Electrocorticography
EEG	Electroencephalography
EM	Eigenvector Methods
ER	emergency room
ERP	Even Related Potential
F	frontal

FFT	Fast Fourier Transform
fMRI	Functional Magnetic Resonance Imaging
FN	False negative
fNIRS	functional Near-InfraredSpectroscopy
FP	frontal pre
fUS	Focused ultrasound-mediated suppression
GA	Genetic Algorithm
HSL	hue, saturation, lightness
HSV	hue, saturation, value
Hz	hertz
IBM	International Business Machines Corporation
ICU	Intensive Care Unit
IL	Inductive Logic
ILP	Inductive Logic Programming
IP	image processing
KDD	Knowledge Discovery in Databases
KNN	K-Nearest Neighbor
LCS	learning classifier
MDP	Markov decision process
MEG	magnetoencephalography
ML	Machine learning
MLP	multi-layered participant
MM	Mathematical morphology
MNIST	Modified National Institute of Standards and Technology
MRI	magnetic resonance imaging.
MRS	Magnetic resonance spectroscopy
NIRS	Near-Infrared Spectroscopy

NMR	Nuclear Magnetic Resonance
OCSP	Oxford shire Community Stroke Project
PCA	Principle Component Analysis
PET	positron emission tomography
PKD	Practice of Knowledge Discovery in Databases
PKDD	Practice of Knowledge Discovery in Databases
PLS	Partial Least Squares
POS	Point-of-sale
PSD	power spectral density
PSO	particle swarm optimization
qEEG	quantitative electroencephalogram
REST	Reference Electrode Standardization Technique
RF	Random Forest
RGB	Red Green Blue
ROC	receiver operating characteristic
SIANN	space invariant artificial neural networks.
SPECT	Single-Photon Emission Computed Tomography
SVM	support vector machine
TFD	Time Frequency Distributions
TIA	Transition Ischemic Attack
TN	True negative
TP	true positive
UCI	University of California Irvine
WT	Wavelet transform

The list of symbols

α	EEG Alpha Band
β	EEG Beta Band
γ	EEG Gamma Band
δ	EEG Delta Band
Θ	The unit step function
θ	EEG Theta Band
σ	Activation Function
Φm	Approximate Entropy
µI	the Mean
ƜI	the probabilities of classes
σ2	variance
std(X)	standard deviation
cov(X,Y)	covariance

Chapter One

Introduction

Chapter One

Introduction

1.1 Preface

The primary focus of research into stroke diseases is prevention via early identification and treatment, with the ultimate aim of preventing a person's mortality after a stroke has already occurred**.**

A stroke is a brain attack. It happens when the blood supply to part of your brain is cut off. Blood carries essential nutrients and oxygen to your brain. Without blood, your brain cells can be damaged or destroyed and they won’t be able to do their job. Work outpaced strokes. Two methods are used in this theses. Electroencephalography should be used first for cranial signals. Next, use industry-specific algorithms to perform further study[1].

The second path is that image processing may also modify signals. Use several methods to enhance signals and achieve the best outcomes. This study focuses on stroke categorization and prediction strategies the s to avoid strokes. All studies reach the same result. stroke's high fatality rate [1]

Because of the nature of the task, the initial point of entry into the study is a set of data groups that each represent a large number of people who have had strokes. These data groups are captured in great detail using electroencephalography (EEG) equipment. The recording period should be relatively close to all of the participants, and careful attention should also be given to the number of channels that are used throughout the recording process. Other factors, such as the frequency range of 12 to 250 Hz, were also taken into account.

Another consideration within the scope of this endeavor is used. Machine learning (ML) is a field of study devoted to understanding and developing methods that 'learn', or methods that use data to enhance performance on a set of tasks . It is considered a component of artificial intelligence.

The premise upon which learning algorithms are constructed is that prior successes indicate that similar approaches, algorithms, and assessments will provide similar results in the future. Machine learning success depends on having a lot of data. Data is initially divided into training and testing parts. Depending on the application, it can then use a variety of algorithms. Most focus on categorization and prediction. Algorithms depend on requirements. In the below of block diagram Figure 1.1, the operation of the work is clear, can divided with two paths first, **the time series** that represent by the electroencephalography **(EEG)** to extract feature (attribute), in that bath used machine learning way to complete the operation.

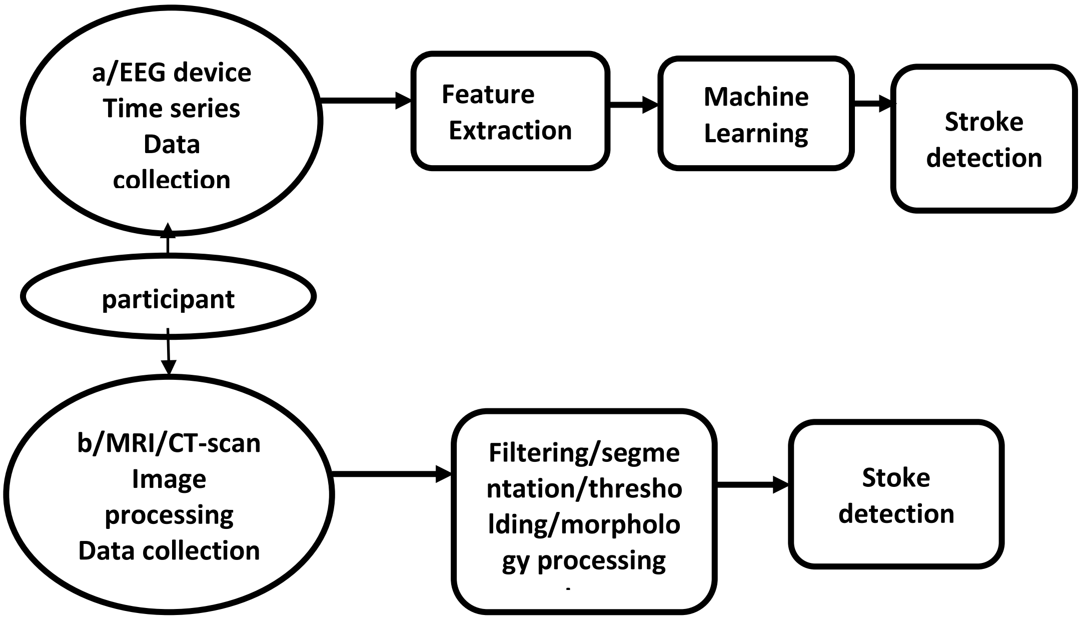


Figure 1.1 The Block Diagram of The Dissertation

The second path refers to image processing, used exactly magnetic resonance imaging. (MRI), and some enhancement operation that extracts the feature from the collection data.

From the block diagram, the important step represents the data collection from any participant and maybe used more techniques exactly in the image processing path, like **positron emission tomography pet-scan, pet-ct, pet-MRI,** computerized tomography **(CT**) and other types of new technology,

Multiple algorithms have been employed in our study based on their actual categorization and prediction principles. The objective is to classify and predict data by developing and evaluating many algorithms and picking the most promising one.

1.2 several key algorithms in this work

A multilayer perceptron (MLP)

MLP is a fully connected class of feedforward artificial neural networks (ANN). The term MLP is used ambiguously, sometimes loosely to mean *any* feedforward ANN, sometimes strictly to refer to networks composed of multiple layers of perceptions (with threshold activation), Multilayer perceptron are sometimes colloquially referred to as "vanilla" neural networks, especially when they have a single hidden layer [2].

An MLP consists of at least three layers of nodes: an input layer, a hidden layer, and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training [3]. It's multiple layers and non-linear activation distinguishes MLP from a linear perceptron. It can distinguish data that is not linearly separable [4].

- **Decision Tree**

It is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements. Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal, but are also a popular tool in machine learning [5] .

- **Random forests**

An ensemble learning approach for classification, regression, and other problems, random forests create several decision trees at training time. The random forest classifies using the majority of trees. Regression tasks yield the trees' mean prediction. Random decision forests remedy decision trees' overfitting to their training set. Random forests outperform decision trees, while gradient-boosted trees are more accurate. Data qualities may impact performance.

The first algorithm for random decision forests was created in 1995 by Tin Kam Ho [6]. Using the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.

Leo Bierman and Adele Cutler trademarked "Random Forests" in 2006 (owned by Minitab, Inc.) .The modification uses Bierman's "bagging" notion and random feature selection, initially presented by Ho and then separately by Amit and German, to create decision trees with controlled variance [7].

- **Extra Trees Classifier**

A decision tree-based ensemble learning approach. Like Random Forest, Extra Trees Classifier randomizes input and choices to avoid overlearning and overfitting. High-variance ensemble methods will precede Extra Tree's Classifier.. Extra Trees, like Random Forest, generates several trees and divides nodes using random subsets of features, but it does not bootstrap observations (meaning it samples without replacement) and splits nodes on random splits, not optimal splits. Extra Trees [8].

- builds multiple trees with bootstrap = False by default, which means it samples without replacement
- nodes are split based on random splits among a random subset of the features selected at every node.

1.3 Stroke in the Human

A stroke may impair bodily function since the brain is responsible for it everything. Stroke, or a cerebrovascular event as it is more formally defined in the medical community, is distinguished by its sudden onset and severe physical consequences. There is usually some underlying cause for a stroke, hence the right phrase to use is "stroke" [8].

A stroke is sudden and the effects on your body are immediate CVA – this stands for cerebrovascular accident (the medical name for a stroke). It is better to say ‘stroke’ as strokes are not accidents – there is always a cause [9].

The term "infarction" refers to the damage caused when brain tissue stops receiving its usual flow of blood.

Blood flow to the brain can be cut off by:

- a blockage (ischemic stroke), or
- a bleed (hemorrhagic stroke).

There are two main types of stroke. The most common type of stroke (about 85% of cases) is caused by a blockage. This is called an ischemic stroke and may be caused when:

a blood clot forms in a main artery to the brain (sometimes called a cerebral thrombosis)

- *a blockage, usually* a blood clot from the heart, is carried in the bloodstream to one of the arteries supplying the brain (called a cerebral embolism), or

- *a blockage forms* in the tiny blood vessels deep within the brain (called a lacunar stroke).

Less commonly, (about 15% of cases) strokes are caused by bleeding in or around the brain. This type of stroke is called a hemorrhagic stroke. It may be caused when:

A blood vessel bursts within the brain (an intracerebral hemorrhage), or a blood vessel on the surface of the brain bursts, causing bleeding into the area between the brain and the skull (called a subarachnoid hemorrhage) [9].

1.3.1 Transient ischemic attack (TIA) definition

TIAs occur when cerebral blood flow is disrupted briefly. A mini stroke. The symptoms are comparable to a stroke (weakness on one side, speech issues, and vision loss), but typically only last a few minutes or hours. Guaranteed 24-hour recovery. During a TIA, brain briefly runs out of oxygen.

A TIA indicates that this portion of your brain is not receiving enough blood and may lead to a more severe stroke. TIA symptoms should be treated immediately. Don't wait for a TIA to resolve acute stroke-like symptoms. Even if the symptoms improve, get medical attention since it might be a stroke [9].

1.4 Literature survey

Cerebral stroke detection research was reviewed. Strokes occur. Ischemic stroke mortality prevention was studied. This stroke may heal. But hemorrhage This review compiles stroke detection and treatment studies to save lives.

Electroencephalography EEG detected strokes initially. Four AIS signals (c3, c4, o1, o2) indicate a stroke. MRIs clarified. Python machine learning and EEG create MLP, random forest, decision tree, and additional tree algorithms. This study correctly identifies stroke patients using machine learning and deep learning.

J. N. Fink *et al.*, (2002) [10], introduced sleeping stroke victims don't know when the strokes rated. Functional imaging might predict acute stroke therapy risks and benefits. Clinical and multimodal MRI outcomes were compared between individuals with known and unknown stroke onset. A prospective stroke registry enrolled individuals who were imaged within 24 hours of an ischemic stroke between January 1997 and June 2000. Clinical and imaging data were compared between group I (established stroke onset) and group II (woke up with stroke).

R. C. Hwa et al., (2004) [11], introduced In order to detect strokes earlier, this article suggests analyzing EEG time series. Detrended fluctuation analysis identifies two channel scaling zones. Using deployed geodesic sensor networks, each subject has up to 128 scaling exponents. Stroke index S is calculated from these scaling exponents' standard deviations. S=1.3 separates our sample of 28 healthy adults from our stroke group. We show that stroke damage to EEG signals is widespread, unlike MRI damage.

R. S. Jeena et al., (2013)[12], performed two-path MRI and CT scans cure strokes.. Multidimensional imaging is essential for diagnostic radiology. CT and MRI provide three-dimensional images. CT brain scans frequently precede stroke diagnosis. MRIs characterize soft tissue well. This article contrasts CT and MRI stroke images. For brain infarct and hemorrhage identification, use digital image processing. Medical photos are median filtered. Gabor filtering and region growth segment. Brain infarct CT and MRI images demonstrate the process. Observing method outcomes. MRI imaging outperforms CT imaging for stroke detection.

B. J. Kim *et al.*, (2014) [13], used in that study, Multimodal MRI may help diagnose stroke, assess thrombolysis risks, and predict outcomes. Due to its high sensitivity and specificity, diffusion-weighted imaging (DWI) can identify acute ischemic stroke from stroke imitators. The lesion mismatch between PWI and

DWI may indicate salvageable tissue by reperfusion treatment. The best threshold for distinguishing benign oligemic regions from the penumbra is currently debated. Fluid-attenuated inversion recovery image signal variations in DWI lesions may reflect ischemic lesion age and hemorrhage risk following thrombolysis. Clots on gradient echo images may indicate their type, and their position, length, and shape may predict reperfusion treatment recanalization.

A. S. Al-Fahoum et al., (2014) [14], introduced pattern section attributes, dimensions, and functions. Extracted features retain signal information. They reduce data explanation resources. Data compression, implementation complexity, and processing costs decrease. Recent EEG feature extraction methods include time frequency distributions (TFD), fast Fourier transform (FFT), eigenvector methods (EM), WT, and auto regression method ARM. EEG signal processing objectively captures brain activity for medical diagnostics and rehabilitative engineering brain-computer interface research. This study will assess traditional EEG feature extraction techniques for certain activities and propose the best way.

S. K. Wijaya et al., (2015) [15], used regular EEG is often used to diagnose brain circulation diseases like strokes. EEGs detect epilepsy, seizures, Alzheimer's, cerebral dysfunctions, and brain degradation. Software enhanced type C hospital stroke EEGs. EEG and CT-Scan were done 48 hours later. Spectral analysis of each channel, Welch analysis for frequency drop, BSI (brain symmetry index) for left and right hemispheres, and other clinical criteria examined EEG data. These signals matched CT abnormalities. CT-scans differed from EEGs. CT-scanned stroke sufferers. Epileptic, stroke, and normal EEGs. Since their BSI levels were above healthy people (0.042 ±0.005), all patients had Acute Ischemic Stroke. 20% had normal EEGs. All topics were unusually powerful. More patients are needed to corroborate these findings.

H. J. Audebert et al., (2015) [16],suggested in cases of acute stroke, imaging enables the identification of subtype, tissue perfusion, and vessel patency. Recent clinical investigations that are likely to inform therapeutic decisions are highlighted in this article. Clot length in computed tomography (CT) and clot burden in magnetic resonance (MR), imaging of leptomeningeal collaterals, and active hemorrhage indicators are depicted. Imaging-based concepts for stroke treatment upon awakening and pre-hospital care in specialized ambulances offer new opportunities to enhance patient outcomes.

W. R. W. Omar et al., (2015) [17], analyzed This article an ischemic stroke patient's EEG signal. The EEG signal was acquired from two channels, Ch. A and Ch. B, using a band pass filter with a frequency range of 0–30 Hz. The signal was FFT-analyzed. EEG data showed individual brainwave patterns across all frequency ranges. EEG data analysis helps understand brain electrical activity and its physiological importance.

J. Wu et al., (2016) [18],sugested EEG has been used to research acute stroke for decades, but its limitations prevent it from informing therapeutic decision-making. EEG hardware, recording electrodes, and software have improved. A dense-array (256 electrodes) EEG, obtained with a saline-lead net and processed using whole brain partial least squares (PLS) modeling, was used to assess acute stroke behavioral impairments and acute brain damage. 24 acute ischemic stroke patients had 3 min of resting-state EEG at bedside in the ER and ICU.

P. Sivakumar et al., (2017) [19],recognized Strokes and brain tumors kill globally. Strokes result from faulty brain cells and blood flow. adaptive neuro-fuzzy inference system (ANFIS) identifies and segments brain MRI images of ischemic stroke. Preprocessing, feature extraction, and classification. Heuristic histogram equalization improves brain imaging. Preprocessing recovers texture and morphology. ANFIS classifies and genetic algorithm GA optimizes. The suggested ischemic stroke detection system's sensitivity, specificity, accuracy, positive and negative predictive values, and Mathew's correlation coefficient are assessed.

P. Vilela et al., (2017) [20], used imaging detects ischemic stroke. CT/MR imaging eliminates stroke mimics, hemorrhage, etiology, mechanism, brain infarct extension, and vascular obstruction. Imaging may identify patients who might benefit more from revascularization outside of the normal treatment timeframe, allowing tailored therapy and better patient outcomes. CT perfusion, diffusion-weighted, and MR imaging may reveal irreversible brain damage penumbra and core. Imaging determines clot kind and extent. Image, therapeutic, and patient results are evaluated.

A. Subudhi et al., (2018) [21], recognized Ischemic stroke detection by brain MRI is crucial and difficult. Optimize brain Diffusion-weighted magnetic resonance imaging (DWI MRI) stroke lesion detection. The approach was evaluated in a slice with a big stroke region using 292 real-time pictures from IMS and SUM Hospital stroke sufferers. Pre-processing, segmentation, feature extraction, and stroke classification are advised. particle swarm optimization (PSO) stroke lesions. The Oxford shire Community Stroke Project (OCSP) used support vector machine (SVM) classifier to classify the feature set into three stroke categories after extracting significant features using the gray-level co-occurrence matrix (GLCM) technique. Darwinian particle swarm optimization (DPSO) algorithms segmented lesions better than PSO (85.19% vs. 90.23%).

A. Qureshi et al., (2018) [22],introduced to Stroke kills second most Americans. Ischemic strokes account for 87%. MRI is the gold standard for identifying ischemic strokes, but 24/7 surveillance is extremely time-consuming. Multi-domain EEG brain signal analysis employing wearable EEG sensors and machine learning detects ischemic stroke. MLP and Bootstrap models (Extra-Tree and Decision-Tree) achieve 95% test accuracy with an area under the ROC curve 0.85 using 40 healthy and 40 patient data.

P. Garg et al., (2019) [23], used Electroencephalography (EEG) detects stroke non-invasively and affordably in this study. Brain cells die without oxygen and nutrients when blood supply is cut off. MRIs or CTs detect brain strokes. We wanted a cheaper, effective CT/MRI option. Thus, stroke patients' 48-hour CT/MRI and EEG data were gathered. CT-scan/MRI images are compared to brain electrical current density. Ischemic and hemorrhagic stroke Cross-sectional investigations show cortical electrical current density. All CT/MRIs showed hemorrhagic or ischemic stroke.

L. Shreve *et al.*, (2019) [24], discovered Early stroke diagnosis improves reperfusion, although behavioral assessments vary. EEGs may detect cerebral ischemia quickly. This pilot study explored EEG for large acute ischemic stroke ED detection. Methods: A dense-array (256-lead) device recorded 3-minute resting EEGs in US Comprehensive Stroke Center ED patients with suspected acute stroke. Results: 14 acute cerebral ischemia patients (five with severe stroke) and 10 without received EEGs. From stroke to EEG, median time was 6.6 hours; from ED arrival, 1.9. Delta band power (P =.004) and the alpha/delta frequency band ratio (P =.0006) distinguished major acute ischemic stroke patients (n = 5) from all other suspected stroke patients (n = 19). Hemisphere signals were diagnostically superior.

S. Bhattacharjee et al., (2019) [25], used the EEG and MRI to understand stroke. EEGs monitor brain electrical activity and may identify neurological diseases. This study analyzes many EEG and MRI **approaches** for brain disease diagnosis. EEG and MRI are compared extensively. This study has two sections.

The first analyzes EEG processing. Next, EEG vs. MRI for brain neurology diagnosis.

Y.-A. Choi et al., (2021) [26], recognized Stroke ranks third in worldwide mortality behind cancer and heart disease. Stroke rates will have quadrupled due to aging by the year 2030. Since chronic illness is the top three worldwide killer, healthcare is crucial. Real-time health prediction algorithms are gaining popularity. MRI diagnoses and prognosticates most elderly strokes. MRI's lengthy testing and high costs make stroke diagnosis and prediction difficult. This work develops and deploys a health monitoring system to anticipate elder stroke precursors while walking. FFT preprocessed six-channel EEG data. The raw spectra yielded alpha (α), beta (β), gamma (γ), delta (δ), and theta (θ) EEG power values, as well as the low β, high β, and β-to-β ratio.

M. Kaur et al., (2022) [27], Advanced Technology aids healthcare infrastructure noninvasively. Stroke, one of the four main cardiovascular diseases, may kill if untreated. Studies show that everyone has TIAs before a stroke. Most cardiovascular disease research uses pricey MRI and CT scan pictures. India requires low-cost, non-invasive stroke detection technologies. This prompted our published research. Authors suggest noninvasive stroke detection. Cascaded prediction algorithms are sluggish, useless on raw data, and don't use EEG's unique qualities.,

A. Sawan et al., (2022) [28], developed in the Computers now dominate health care. decision support systems (DSS) aids patient diagnosis and vital sign readings. Wearable medical gadgets can better record brain impulses than EEG equipment. Mental health and brain analysis need EEG data to detect nerve-related illnesses like stroke. Machine learning is used to diagnose strokes using MUSE 2 EEG ,With Muse's brain sensing headbands[29], data. Using eight ML methods to distinguish strokes, XGboost classifiers outperformed others with

83.89 percent accuracy. Accuracy improved 7.89%. A 3-minute Muse is a device sensing portable EEG predicts stroke severity.

E. Dritsas et al., (2022)[30], occurred Stroke happens when blood supply to the brain suddenly stops. Without blood, brain cells die, causing impairment in many different parts of the brain. Having a stroke detected early on may motivate people to lead healthier lives. In this research, we create and assess ML models to forecast future risks of stroke. This stacking method enhances AUC, Area Under the ROC Curve (AUC) precision, recall, F-measure, and accuracy. The AUC, F-measure, precision, recall, and accuracy were all highest using the stacking classification method.

1.4.1 Summary of the Survey

The results of the aforementioned polls allow us to make certain inferences. Ischemic stroke, a subtype of stroke illness, was our primary emphasis here. There are essentially two ways to go about this: a time series approach or an image processing approach. Depending on the state of the art of the device used to collect the data (for example, a CT scan, a MRI scan, a pet scan, or any number of other developments), the image processing approach is generally preferred. Here, we zeroed in on ischemic stroke, a subtype of stroke, Experts universally favored time series analysis and image processing to treat it. Imaging technologies like CT and MRI scanners and image processing algorithms were becoming more advanced.

Here illustrated the table that include all the above survey. Shown table 1.1.

I had a look at some related issues to our study here.

Table1.1 Illustrated the Summary of The Literature Survey

Ref.	Technique used	Results with conclusion
[9]	**Using imaging** processing techniques to extract the feature	100 (27%) of 364 patients awoke with a stroke. Group I and II had the same age, gender, National Institutes of Health Stroke Scale, and TOAST (Trial of Org 10172 in Acute Stroke Treatment) diagnosis. Group I had a shorter stroke onset (mean 6.0 vs 13.3 hours, $P<0.001$) and a similar detection time (6.0 against 5.9 hours). 82% of group I and 73% of group II patients had DWI-PWI mismatch within 3 hours.
[10]	Using the **Electroencephalography** **As a basic technique to collect data**	Selecting a representative sample of volunteers with the relevant demographic data and other factors is important for data quality. A laying posture reduces artefacts produced by faint motion, but it may also induce sleep, particularly in a darkened, noiseless area. The volunteers should have the same circumstances and instructions throughout the trial.
[11]	Used the electroencephalography **EEG TO extract the data**	EEG time series detection of stroke is proposed. Detrended fluctuation analysis shows two scaling areas per channel. Each topic has 128 scaling exponents using geodesic sensor nets. The scaling exponents' normalized variances yield a stroke index S. S=1.3 distinguishes our 28 normal and stroke individuals. We demonstrate that stroke affects EEG signals globally, unlike radiological investigations like MRI.
[12]	Collect data by **Image technics CT+MRI**	The program uses digital image processing to identify brain infarcts and hemorrhages. Median filtering preprocesses medical pictures. Gabor filtering and seeded region expanding segmentation. CT and MRI brain scans with various infarcts exemplify the procedure. Visually assessing technique outcomes. The suggested stroke detection approach shows that MRI imaging outperforms CT imaging.
[13]	Used image processing path to collect data. MRI + Clinical information	Clinical presentation determines stroke diagnosis. 19%-30% of suspected strokes are stroke-mimics. A patient with a localized neurological impairment requires a comprehensive differential diagnosis. DWI (88%-100% sensitivity and 95%-100% specificity) may now accurately identify ischemia lesions in MRIs. Even 3 minutes after stroke start, DWI and ADC maps show hyperintense and hypointense lesions. MRI may also identify acute cortical or subcortical lesions, notably in the posterior fossa or brain stem, better than CT. However, early DWI may miss brain stem lesions with modest symptoms, such as ataxic hemiparesis or intranuclear ophthalmoplegia.
[14]	Collect data using the EEG. Technique**,**	Five well-known frequency domain and time-frequency domain approaches were described. It is difficult to rank approaches by capacity. The results show that each strategy has pros and cons that suit certain signals. EEG signals may perform poorly using frequency domain algorithms. Frequency domain approaches may give more EEG analysis information than time-frequency methods. When

Ref.	**Technique used**	**Results with conclusion**
		discussing technique performance, the signal to be evaluated must be specified. Thus, each application may need a distinct approach.
[15]	Collect data by **the EEG, AND CT-scan**	This research developed a software tool to improve type C hospital stroke EEGs. EEG and CT-Scan were performed 48 hours after commencement. Spectral analysis, Welch analysis, and BSI (brain symmetry index) were used to evaluate the EEG data. These signals were matched to the CT-scan for anomalies.
[16]	Collect data by imaging **used.** **The MRI+ CT SCAN**	Imaging shows stroke subtype, tissue perfusion, and vascular patency in acute stroke. This review discusses current clinical trials that may inform treatment. CT and MR clot length and load, leptomeningeal collateral imaging, and current bleeding markers are shown.
[17]	Data collection using THE **EEG technique** used 2 channels only	This research analyzes ischemic stroke patient EEG signals. EEG signals from channels A and B were filtered using a band pass filter from 0 to 30 Hz. FFT analyzed the signal. EEG readings indicated pattern variance in each brainwave sub band for each individual. EEG processing helps comprehend brain processes and find their physiological relevance.
[18]	Collect data using the imaging technique (**MRI_ adaptive neural fuzzy interface)**	Brain tumors and strokes are global killers. Brain strokes result from brain cell abnormalities and blood flow restriction. Adaptive Neuro Fuzzy Inference (ANFIS) classifier is used to identify and segment ischemic stroke in brain MRI images. Preprocessing, feature extraction, and classification comprise the approach.
[19]	Collect data using **image processing (CT+MRI)** ------- ------ ----------	Brain tumors and brain strokes kill people worldwide. Brain stroke is caused by brain cell abnormalities and blood flow restriction. This study proposes a computer-aided automated technique to identify and segment ischemic stroke in brain MRI images using Adaptive Neuro Fuzzy Inference (ANFIS) classifier. The suggested technique includes preprocessing, feature extraction, and classification. The brain picture is improved using Heuristic histogram equalization. Texture and
[20]	Collect the data using i**maging** (MRI)	Ischemic stroke detection utilizing brain MRI images is crucial and difficult in clinical practice. Optimization-based stroke lesion detection in brain diffusion-weighted imaging (DWI) MRI sequences is proposed. The system was evaluated in a slice with a big stroke region using 292 real-time pictures from IMS and SUM Hospital stroke patients. Pre-processing, segmentation, feature extraction, and stroke classification are the suggested steps
[21]	Collect data using the **EEG technique**	an ischemic stroke detection approach using multi-domain EEG brain signal analysis using wearable EEG devices and machine learning. Using 40 healthy and 40 patient data, Multi-Layered Perceptron (MLP) and Bootstrap models (Extra-Tree and Decision-Tree) achieve 95% test accuracy with an area under the ROC curve 0.8

Ref.	Technique used	Results with conclusion
[22]	Collect data using. **Electroencephalography (EEG)+ CT-scan/MRI**	This work utilizes EEG to detect stroke non-invasively and cheaply. Brain cells collapse when blood supply to the brain is disrupted, depriving them of oxygen and nutrition. CT or MRI scans are used to locate a brain stroke. CT-scan/MRI is expensive; thus, we sought a cheaper, effective alternative. Thus, stroke patients' CT-scan/MRI and EEG data were collected within 48 hours
[23]	Collect data using the **EEG technique**	EEG measurements indicate significant acute ischemic stroke and correlate with infarct volume within hours after stroke start. These findings imply that EEG brain function parameters may enhance ED diagnosis of massive acute ischemic stroke, which may be relevant for pre-hospital applications.
[24]	Collect data using **EEG +MRI**	EEG records reveal brain disorders. This article details EEG and MRI applications for brain illness detection. EEG-MRI comparisons are also conducted. EEG processing was discussed in the first portion of this article. EEG and MRI brain illness detection comparisons are next.
[25]	Collect Data Using the **EEG Technique**	In this study, we create and deploy a health monitoring system that predicts senior stroke precursors while daily walking. FFT preprocessed six-channel electroencephalography (EEG) data. The raw EEG power values for alpha (α), beta (β), gamma (γ), delta (δ), and theta (θ) were recovered from the raw spectra. This article shows that EEG biometric signals alone while walking may reliably predict senior stroke precursors and incidence with over 90% accuracy.
[26]	Collect Data Using the **EEG Technique**	New technologies enable noninvasive health-care systems. Stroke is the most deadly of the four main cardiovascular disorders. This article proposes EEG-based stroke prediction techniques. This study proposes time series-based techniques including LSTM, biLSTM, GRU, and FFNN to produce effective predictions.
[27]	Collect data using the **EEG technique (wearable device**), machine learning algorithms was used	EEG signals may identify several nerve-related disorders, including stroke, making them important for mental health and brain analysis. Machine learning is used to diagnose strokes using EEG information from the wearable gadget "MUSE 2." Eight ML methods were tried to classify strokes, however the XGboost classifiers had an 83.89% accuracy rate. Accuracy improved 7.89% over the prior research. “A 3-minute Muse portable EEG recording predicts stroke severity.
[29]	Data collected from The **CT SCAN And MRI**	This study develops and evaluates different machine learning (ML) models to provide a viable framework for long-term stroke risk prediction. This paper presents a stacking strategy that performs well in AUC, precision, recall, F-measure, and accuracy. Stacking classification outperformed the other approaches with an AUC of 98.9%, F-measure, precision and recall of 97.4%, and accuracy of 98%.

1.5 Statement of the Problem

The following is a list of some of the challenges that are faced in the area *of* ***Ischemic stroke*** prediction and EEG data processing:

1) The non-linear character of the EEG signal made analysis and information extraction challenging. The patient undergoes a lot of ambient stimulation while being examined to record these signals, which must be considered.
2) A stroke that induces unanticipated seizures causes tremendous distress for the patient and his or her close associates. Consequently, the patient must possess the ability to anticipate the onset of epileptic seizures for a significant duration so that measures can be taken to alleviate their symptoms or completely prevent them through the administration of medication. This will significantly benefit the patient in terms of time management.
3) Thirdly, the lack of physician authorization is a critical issue for most stroke prediction systems. The reason for this might be because they have not been tested on a varied population of patients or that they have a high rate of false alarms. So, the goal is to create a general algorithm, but then modify its parameters to match the patient's EEG data, making it a personalised algorithm that might be approved by the medical community.
4) Furthermore, while the majority of current algorithms utilize multi-channel EEG, the objective of this research is to minimize the channel count to enhance the algorithm's practicality in real-life scenarios.

1.6 The Aims and Requirements of the Work

The development of a stroke prediction system employing image processing and EEG data is the primary objective of this inquiry. The recommendation system in chapter three was centered around this aim, and the additional reasons are listed below.

1) Make an application that can adjust EEG signals and remove artefacts, even without a second artefact channel for projection.
2) It is recommended to propose and evaluate machine learning models for stroke prediction.
3) Suggest for an improved method for multi-stage stroke prediction that is based on feature extraction and machine learning.
4) It is recommended to create a programme that can anticipate strokes in real time. In order to detect an impending stroke, this programmed would examine the periodogram of incoming EEG segments and use the rapid frequency changes across bands as signs.
5) In the image processing used many image enhancement to improve that path, used more advance technique method to achieve the target
6) Sixthly, the IP route must clearly identify the stroke site from other points, such as tumours, lesions, cysts, and lipids, so that the suggested approach can fix this point and provide excellent results.

1.7 Contribution in this study

The following are the primary contributions made by this work:

1) The first step is to extract the characteristics from the signals by making use.
2) of the two different kinds of features, time domain and frequency domain.

3) In this study, uses two different approaches, namely time series and image processing, and obtained findings from each of these approaches, which then compared with one another.
4) In this research, a variety of algorithms may help us achieve our goals and provide better outcomes.
5) The stroke prediction algorithm is a machine learning method, and it consists of four algorithms. The first step is the selection of the EEG channels, and the second stage is the determination of the optimal Preictal interval ***length*** (***PIL***) and segment length.
6) In the EEG path the challenge that how to select the channel position place, and the how many number of channel, these selection according to effect directly to the result, in this study we select four channel only, this choice help any researcher to reduce the collection data and go forward to the target easily
7) For image processing (**IP)** making a new approach to used more enhancement to improvement the result and this can be shown in the chapter three in details
8) The suggested (IP) approach is more flexible and can simply be enhanced to improve results.

1.8 The System's Proposed Restrictions

1) Training the model following its proposal and building requires time and repetition for each participant, making the procedure long. Another issue is hardware, which requires more powerful computers with greater memory to capture more data.
2) The difficulty in collecting high-quality data due to the large number of participants is a third issue. at order to collect reliable data, it is necessary to have the same groups of individuals follow a predetermined protocol at a healthcare facility.

3) Thirdly, it needs to recruit more people to take part in the study since the amount of data needed is massive.

1.9 Formulating Dissertation Strategies

This dissertation's remaining chapters are structured as follows:

- *Chapter two*. Explain the general headline to the direction to stroke with EEG and machine learning, image processing, it's a theoretical background to the direction that I followed.
- *Chapter three* the proposed method that suggestion in my work in the two paths EEG, AND image processing (IP)
- *Chapter four,* show the result in two paths the EEG time series, and image processing with discuss the discuss and compare between them.
- *Chapter five.* derive some conclusions and make some suggestions for new concepts that might be employed in future work to use new methods for collecting data from EEG devices and other new devices.

EEG has been used to research acute stroke for decades, but its limitations prevent it from informing therapeutic decision-making. EEG hardware, recording electrodes, and software have improved. A dense-array (256 electrodes) EEG, obtained with a saline-lead net and processed using whole brain partial least squares (PLS) modeling, was used to assess acute stroke behavioral impairments and acute brain damage. 24 acute ischemic stroke patients had 3 min of resting-state EEG at bedside in the ER and ICU.

Chapter TWO

Background of Signal and Image Processing for detecting strokes

Chapter Two

Background of Signal and Image Processing for detecting strokes

2.1 Introduction

In this chapter illustrated some important subjects that depend on the work, as a background theory and give more information of the related subject.

Stroke daisies are commonly spread through the human beings, therefore the prediction and detection techniques become more important in this situation. EEG output signal processing which is recently used for detecting the brain stroke is a kind of Signal processing technique. Image processing techniques are mainly achieved for detecting the stroke. [1]

2.2 Strokes on the body

The brain is like a computer it sends messages around the body to enable the body to do functions. It is made up of two halves, the left hemisphere, and the right hemisphere. Each part of the brain has a specific job to do. In general, the right half of the brain controls the left side of the body and vice versa. Specific areas of the brain control the ability to move, speak and write. Other areas will control memory, emotions and vision [1].

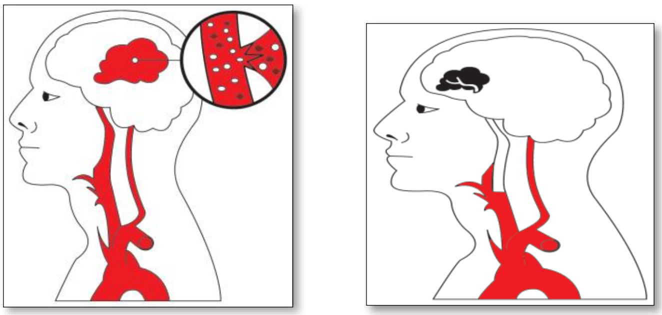
A stroke damages brain cells, so they can no longer work properly. As a result, The regions of the body they regulate are also affected.

In the event of visual impairments, the occurrence of a stroke will result in the impairment of the cerebral region responsible for visual perception. The effects of a stroke vary depending on the part of the brain that is affected.

A stroke damages brain cells so they can no longer work properly. As a result, the regions of the body they regulate are also affected.

In the event of visual impairments, the occurrence of a stroke will result in the impairment of the cerebral region accountable for visual perception.. The effects of a stroke vary according on the part of the brain that was affected.

Figure 2.1(a, b) shows a generic block diagram with two kinds of strokes.



(a) (b)

Figure 2.1 (a) Hemorrhage stroke (b) ischemic stroke

After experiencing a stroke, individuals may have limitations in doing some activities during the first weeks subsequent to the occurrence. While a significant number of persons may have complete rehabilitation, there exists a subset of individuals who may endure a lasting impairment throughout their lifetime [1]

- **Weakness or paralysis**

The most frequent side effects of having a stroke include weakness, clumsiness, and sometimes paralysis. In most cases, they will only affect one side of body. Stiffness (spasticity) of the muscles is a common factor that may make weakness or paralysis of an arm or leg much worse. Language and

verbal communication. In such case, there are other people like. After having a stroke, many people find it hard to talk. Aphasia is the name for this problem.

2.3 Electroencephalography path

EEG bio signals show neocortex and pyramidal neuron postsynaptic potentials [31]. With scalp electroencephalography, the International 10-20 electrodes are used. Intracranial EEG is electroencephalography using surgical electrodes. Clinically, EEG tracings are inspected.

Electrode voltage fluctuations and EEG bio amplifiers assess brain activity. Electrodes on the scalp capture brain tissue neuron activity by direction and distance. Bone and intermediary tissues affect value. Near-scalp electrode cortical neuron activity dominates EEG data. EEGs exclude the base of the cortical gyrus, mesial walls of main lobes, hippocampus, thalamus, and brain stem.

EEGs that are healthy indicate that the subject is awake. 1–30 Hz, 20–100 µV[32]. Alpha, beta, delta, and theta singles should be detected. Alpha waves rule calm wakefulness. Mental effort increases beta waves. Beta activity increases in calm persons. Wakefulness lacks theta and delta waves, suggesting brain dysfunction. [32]

EEG detects acute waves, spikes, and spike-and-wave complexes to diagnose epilepsy. EEG detects seizures, status epilepticus, and spatial-temporal evolution. It detects sleep problems, anesthetic depth, coma, encephalopathies, cerebral hypoxia following cardiac arrest, and brain death. MRI and CT now diagnose cancer, stroke, and other brain problems before EEG. EEG is useful for research and diagnosis despite poor spatial resolution. Its temporal resolution is milliseconds, Figure 2.2 shows the real sample of EEG, and Figure 2.3 shows the standard location to all sensors. [32]

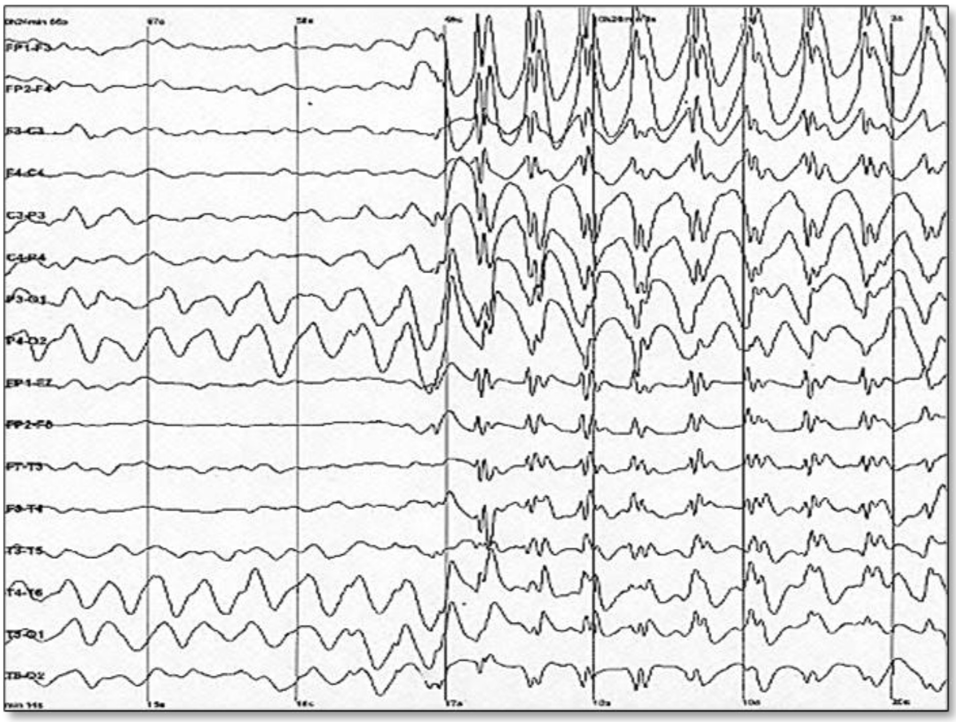


Figure 2.2 An Example of EEG Signals

Two frontal (F_{P1}, F_{P2}) and two occipitals (O_1, O_2) stroke detection signals are used in this research.

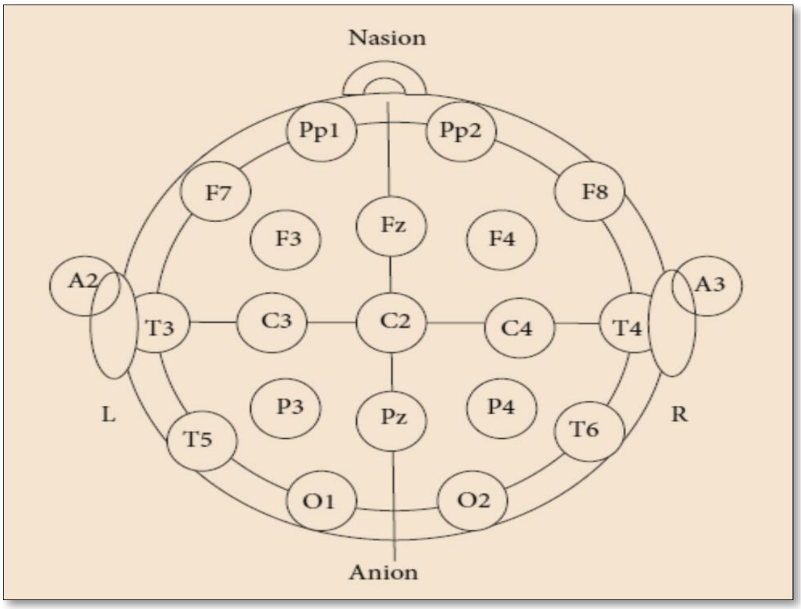


Figure 2.3 standardization location of electrode placement-use standard

2.3.1 Advantages of Electroencephalography (EEG)

Functional magnetic resonance imaging fMRI, positron emission tomography PET, Magnetoencephalography MEG, Nuclear magnetic resonance NMR, Near-infrared spectroscopy NIRS, are other methods for studying brain function. EEG one-dimensional signals from localized peripheral regions on the head make it attractive for its simplistic fidelity and has picture processing and other secondary aims which may be summed up as permitting a high clinical and fundamental research throughput despite its inadequate spatial sensitivity . EEG has advantages over alternative methods.

a) Hardware costs are significantly lower than those of most other techniques.
b) EEG prevents limited availability of technologists to provide immediate care in high traffic hospitals [32].
c) EEG requires a quiet room and briefcase-sized equipment, whereas fMRI, single-photon emission computerized tomography SPECT, PET, MRS, and MEG need bulky, inflexible equipment. MEG employs liquid helium-cooled detectors in magnetically insulated chambers and costs several million dollars, whereas fMRI needs a 1-ton magnet.[32] .

d) EEG has high temporal resolution (although sub-millisecond solution generates less meaningful data) because the two to 32 data streams generated by that number of electrodes are easily stored and processed, whereas 3D spatial technologies provide thousands or millions of input data streams and are limited by hardware and software. Researchers and clinicians record 250–2000 Hz EEG [32]. Motion-tolerant EEG. Reduce EEG movement artifacts.

2.3.2 Disadvantages of Electroencephalography (EEG)

a) Low scalp resolution. EEG is difficult to understand, while fMRI can directly demonstrate brain activity [33]

b) Due to the inverse issue, the dipole generating an EEG change may be mis localized depending on its orientation and location [34]
c) EEG inadequately measures neuronal activity below the cortex.
d) Unlike PET and MRS, cannot locate neurotransmitters, medications, etc. in the brain [35]
e) Requires perfect skull placement of hundreds of electrodes, conductivity gels, saline solutions, or pastes, and a cover. EEG prepares MEG, fMRI, MRS, and SPECT quicker.
f) Requires perfect skull placement of hundreds of electrodes, conductivity gels, saline solutions, or pastes, and a cover. EEG prepares MEG, fMRI, MRS, and SPECT quicker.

Since an EEG voltage signal indicates a difference between two electrode voltages, the reading Encephalographic may be shown in numerous ways.

2.3.3 The representation EEG channel montage

a) Sequence montage. Each channel (waveform) represents the difference between nearby electrodes. These channels comprise the montage. Fp1-F_{P3} the voltage difference between the Fp1 and F_{P3} electrodes. The montage's channel illustrates the electrode array's voltage differential [36].
b) Reference montage. Each channel displays the electrode-reference electrode difference. Unlike the "recording" electrodes, this reference has no fixed position. Online, midline spots like Cz, Oz, Pz, etc. don't boost one hemisphere's signal. Offline references
c) Rest Reference: zero-potential offline computational reference at infinity. REST (reference electrode standardization method) employs equivalent sources within the brain of scalp recordings to connect the real recordings with any online or offline (average, linked ears, etc.)

non-zero reference to the new recordings with infinity zero as the standardized reference.

d) Linked Ears: an average of electrodes on both earlobes or mastoids.

e) Average reference montage. All amplifier outputs are averaged and used as the channel reference.

f) Laplacian montage: Each channel displays the difference between an electrode and a weighted average of its neighbors. The technician changes montages during analog EEG recordings to accentuate or clarify EEG features. Digitized EEG retains all signals in a single (typically referential) montage. Since any montage can be mathematically generated from another, the electroencephalographic may study the EEG in any display montage [37].

A clinical neurophysiologist or neurologist with clinical EEG interpretation expertise reads the EEG. quantitative electroencephalogram (**qEEG**) is controversial for therapeutic use yet commonly used in research.

2.3.4 Artifacts

The signal captured by EEG is constantly polluted by artifacts, which might affect data processing. Non-brain signals are artifacts. Artifact removal algorithms exist, but how to handle them is unclear. Artifacts may come from instrument faults including malfunctioning electrodes, line noise, or excessive electrode impedance, Figure 2.4 shows a real sample of artifacts capture [38].

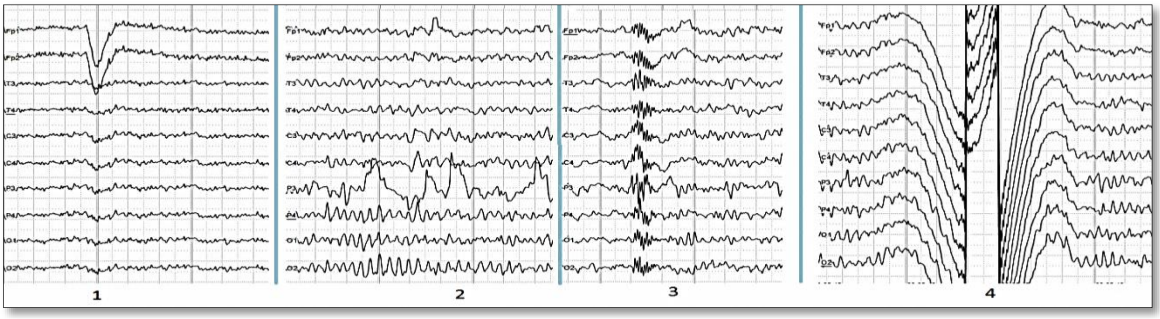


Figure 2.4 Main types of artifacts in humane

2.4 Artificial intelligence overview

Artificial intelligence, also known as machine intelligence, is intelligence that is proven by machines as opposed to natural intelligence, which is demonstrated by people and other animals. This is in contrast to natural intelligence, which is presented by humans and other animals. Speech recognition, learning, strategic planning, and problem solving are only few of the functions that are intended to be carried out by it. Robotics is the branch of study that focuses on the link of perception to action; hence, Artificial Intelligence has to play a pivotal part in robotics for there to be an intelligent relationship between perception and action. Artificial intelligence seeks to answer fundamental concerns such as what kinds of information are necessary for different aspects of thinking, how those kinds of knowledge should be represented, and how those kinds of knowledge should be used. Robotics presents a challenge to artificial intelligence by compelling it to interact with real-world objects and events in their natural environments [39]

2.4.1 short introduction to artificial intelligence

The artificial intelligent (AI) led to machine learning. Early AI researchers intended computers to learn from data. They attempted to solve the issue using symbolic methods and "neural networks." Perceptron and other "neural networks" were revealed as statistical generalized linear models. Computer-assisted medical diagnosis using probability. AI and machine learning may vary due to the rise of knowledge- and logic-based methods. Probabilistic systems have theoretical and practical data collecting and representation challenges. This affected system design and use the concept of the (AI) shows in the Figure 2.5.

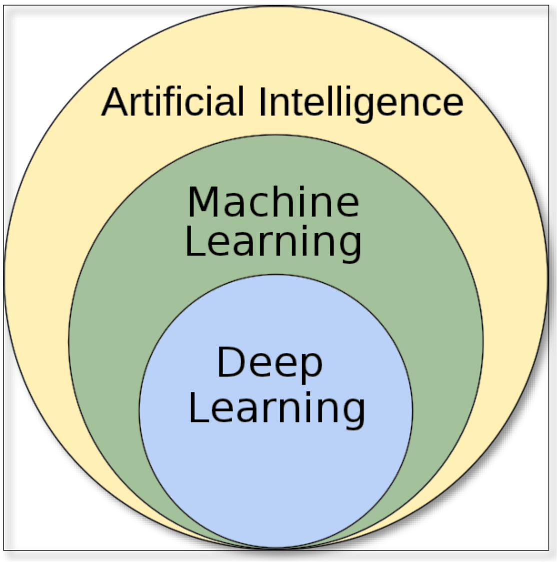


Figure 2.5 Machine learning as subfield

Machine learning (ML) surged in the 1990s. Recently, artificial intelligence has given way to practical problem-solving. Statistics, fuzzy logic, and probability theory superseded symbolic AI. Performance improved. AI and machine learning are confounded. Machine learning (ML) can learn and produce predictions without.

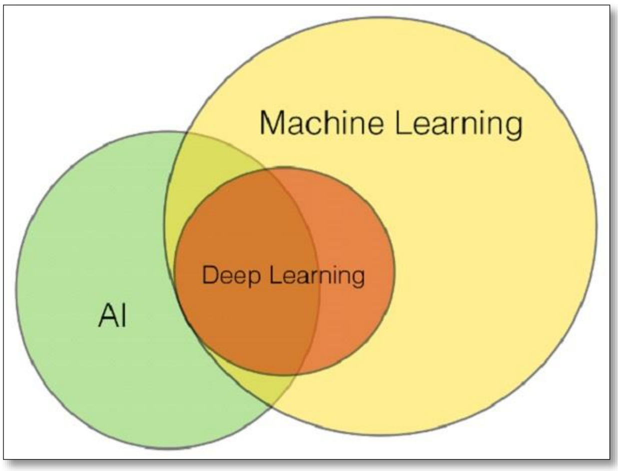


Figure 2.6 Part of machine learning as subfield of AI or part of AI as a subfield of Machine learning

active participation from the user, unlike artificial intelligence (AI). shows in Figure 2.6 [40].

2.4.2 Data mining

Data mining finds new features in data, whereas machine learning predicts based on training data. Data mining employs machine learning for multiple purposes. Data mining for "unsupervised learning" or preprocessing improves machine learning accuracy. **The knowledge discovery** in databases (**KDD**) discovers new information, whereas machine learning reproduces current knowledge to assess performance. These two academic groups often have distinct conferences and publications, such ECML PKDD, and their assumptions lead to misunderstanding. In a typical KDD project, there is not enough training data to apply supervised methods, but an ignorant (unsupervised) approach will quickly be outperformed [41].

2.4.3 Unsupervised learning

Unsupervised learning clusters data points to identify structure. Algorithms learn on unlabeled, uncategorized test data. Unsupervised learning algorithms avoid criticism by finding data commonalities. Ignore input. Unsupervised learning's probability density function application is significant. Statistics uses density estimation. Data summary and interpretation use unsupervised learning. Cluster analysis uses specified criteria to group observations into groups. Cluster observations vary. A similarity metric defines the data structure and evaluates it in several ways, such as internal compactness and separation. Density and graph connectedness underlie other approaches [42].

2.4.4 Semi-supervised learning

Semi-supervised learning is between unsupervised and supervised. Many machine learning researchers have discovered that adding a little amount of labeled data increases learning accuracy. even without labelling. Poorly supervised learning employs noisy, restricted, or incorrect training labels yet is cheaper, resulting in larger effective training sets [43].

2.4.5 Reinforcement learning

Reinforcement learning examines how software agents might optimize cumulative reward. This area covers software agent behavior. Due of its generality, game theory, control theory, operations research, information theory, simulation-based optimization, multi-agent systems, swarm intelligence, genetic algorithms, and statistics study it. Machine learning models the environment as MDP. Many reinforcement learning methods use dynamic programming. Since accurate MDP mathematical models are impossible, reinforcement learning may be utilized without them. Autonomous cars and gaming AI employ reinforcement learning [44].

2.4.6 Self-learning

Self-learning and the crossbar adaptive array (CAA) neural network were introduced in 1982. Education without incentives or teacher guidance. The CAA algorithm for self-learning cross-computes action judgments and consequence attitudes. It is driven by thought-feeling dynamics. Self-learning modifies the memory matrix W = ||w(a,s)|| [45] in such a manner that the subsequent machine learning procedure is carried out at each iteration: Conditions call for intervention. reliability determined by the numbers [45] .

2.4.7 Feature learning

Acquiring features training algorithms increase data representations. Cluster and PCA are classics. Representation learning algorithms, or feature learning algorithms, maintain and utilize input. Categorization and prediction often need pre-processing. It reproduces unknown inputs. Unbiased data distribution. data-generating distribution. This lets a computer learn and apply features to execute a function without feature engineering. Unsupervised or supervised learning works. Supervised feature learning chooses features from labeled input data. AI, multilayer perceptron, and supervised dictionary learning are examples. Unsupervised feature learning learns from untagged data [41].

2.4.8 Models

There are numerous different models for machine learning, and the vast majority of them are constructed on top of certain machine learning algorithms. The more common machine learning algorithms of classification and regression belong to the category of supervised machine learning, while clustering techniques are often used in situations involving unsupervised machine learning.

2.4.9 Artificial neural networks

Artificial neural networks (ANNs), also known as connectionist systems, resemble organic neural networks in animal brains. ANNs are models of linked "artificial neurons." They resemble brain neurons. Each connection may "signal" an artificial neuron, like brain synapses. Artificial neurons analyze and transmit.

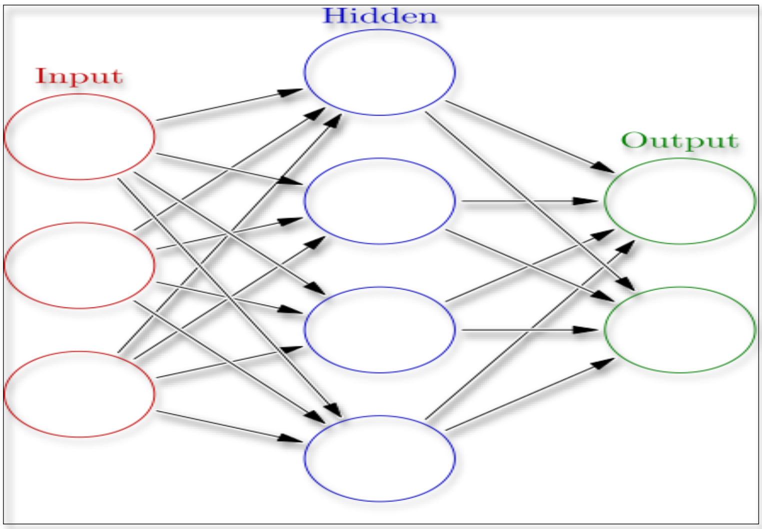


Figure 2.7 An artificial neural network[41]

impulses. In most ANN implementations, each artificial neuron's output is a non-linear function of its inputs, and its link signal is a real integer. Artificial neural networks convey numbers. Edges simulate neurons. Learning alters artificial neuron and edge weight, in Figure 2.7 shows an example to the ANN [41]

AI is used in computer vision, speech recognition, machine translation, social network filtering, board and video games, and medical diagnosis [46].

2.4.10 Decision trees

The use of decision-tree models for making projections. The process involves making observations at the branches and afterwards forming value judgments based on the characteristics of the leaves. Statistics, data mining, and machine learning employ predictive modeling. Classification trees have restricted target variables. These trees have class labels as leaves and characteristics as branches. Regression trees are used for continuous real-number target variables. Decision trees may illustrate indecision. Data mining uses decision trees, although their categorization trees may be used for decision-making Figure 2.8 [46].

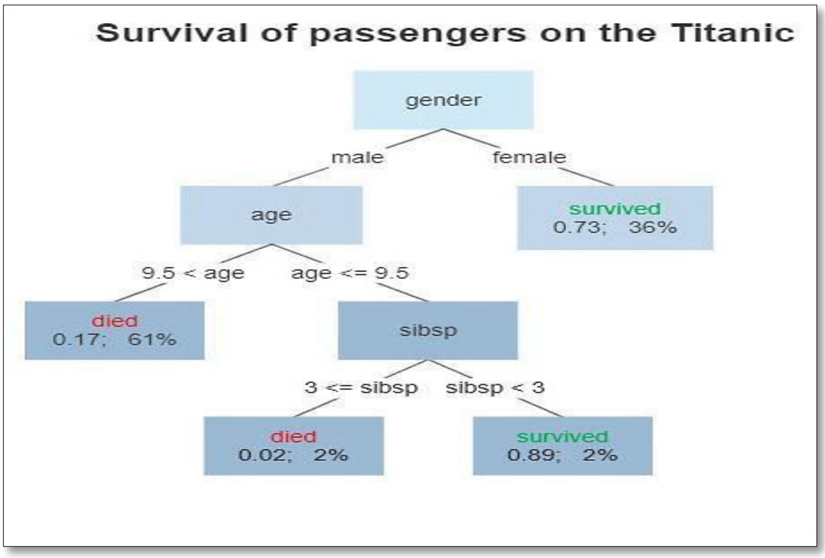


Figure 2.8 simple flow to session tree algorithm[46]

2.4.11 Support-vector machines

Support-vector networks (SVMs) are linked supervised learning methods for classification and regression. An SVM training method builds a model that predicts whether a new sample belongs to one of two categories using a series of training samples. SVMs are binary, linear, non-probabilistic classifiers. Platt scaling may classify probabilistically SVMs may do non-linear classification using the kernel technique [47].

2.4.12 Machine learning's bias

Data biases most impact machine learning. Machine learning systems trained on present customers may not forecast future client demands. Machine learning may learn society's constitutional and unconscious prejudices from human data. Language models show biases. Criminal risk algorithms discriminate against African Americans. 2015 Google Photos misidentified black people as gorillas [48].

2.4.13 Machine Learning Definition

AI implementation determines robots' future. Making robots think is difficult, however. Only machine learning (ML) can produce robust AI. Machine learning enables computers to learn without programming. Machine learning develops data-driven computer algorithms.

2.4.14 The Mechanisms of Machine Learning

Like the brain, machine learning utilizes training data or knowledge graphs to grasp things, domains, and their relationships. Identifying entities starts deep learning. Data gathering starts machine learning. It draws conclusions from instances using data patterns. Machine learning lets computers adapt without human help [45].

2.4.15 The Importance of Machine Learning

Machine learning is ancient. IBM computer scientist Arthur Samuel invented "machine learning." Samuel's software played checkers. Playing more enhanced its game prediction systems. Machine learning creates data-learning algorithms.

Machine learning solves problems quicker and better than people. Machines may automate regular tasks by detecting patterns and correlations in incoming data if supported by vast computing resources.

- *Data Is Key*: ML algorithms are crucial. ML algorithms generate a mathematical model using "training data" to predict or judge without programming. This may reveal data trends organizations may use to improve decision-making, productivity, and actionable data.
- *AI Is the Goal*: ML enables AI systems to automate processes and solve data-based business challenges autonomously. It lets organizations replace **or** enhance human skills. Chatbots, self-driving vehicles, and voice recognition use machine learning**.**

2.4.16 Machine Learning Is Widely Adopted

There is such a thing as ML. There are several options for a flight. due to the epidemic. In 2021, artificial intelligence (AI) was embraced by 41% of businesses. New firms account for 31% of those utilizing or testing AI.

- Data security: ML exists. It boosts creativity and productivity across industries. 41% of companies utilized AI more after 2021's epidemic. These firms join 31% of AI adopters.
- Finance: Banks, trading brokerages, and finch businesses utilize machine learning algorithms to automate trade and advise clients. Erica, a Bank of America chatbot, automates customer care.

- Healthcare: ML analyzes enormous healthcare data sets to expedite therapy discovery, enhance patient outcomes, and automate routine activities to reduce human error. IBM's Watson mines data to help doctors customize patient care.
- Fraud detection: Financial institutions are using AI to detect fraud in real time by analyzing vast quantities of transactions. Capgemini says that machine learning and analytics fraud detection technologies save inquiry time by 70% and increase accuracy by 90%.
- Retail: AI researchers and developers are using ML algorithms to develop AI recommendation engines that offer relevant product suggestions based on buyers' past choices, as well as historical, geographic and demographic data [48]

2.4.17 Introduction to Data in Machine Learning

- Data: It may be anything: a fact, a number, a word, a sound, a picture.. Data is the key to Data Analytics, Machine Learning, and AI. Modern research and automation will fail without data. Big companies spend a lot of money collecting data.
- Information: Data that has been interpreted and manipulated and has now some meaningful inference for the users.
- Knowledge: Combination of inferred information, experiences, learning, and insights. Results in awareness or concept building for an individual or organization. The Figure 2.9 shows the steps of processing.

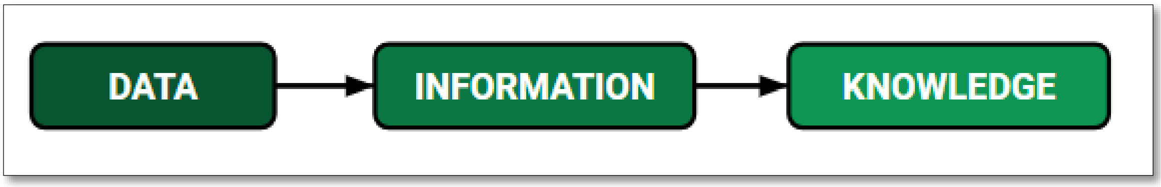


Figure 2.9 data processing steps[48]

2.4.18 Machine Learning data splitting

- Training Data: The part of data use to train our model. This is the information that the model encounters (both input and output).
- Validation Data: The data used to often evaluate the model, fit on the training dataset, and improve hyper parameters (pre-set parameters before the model starts learning). This data aids model training [49].
- Testing Data: Testing data is impartial once our model is trained. Our model
- predicts values from Testing data inputs. After prediction, compare model to testing data output. Its measure our model's learning from the training data in this way [50]. the Figure 2.10 shows the processing of split data.

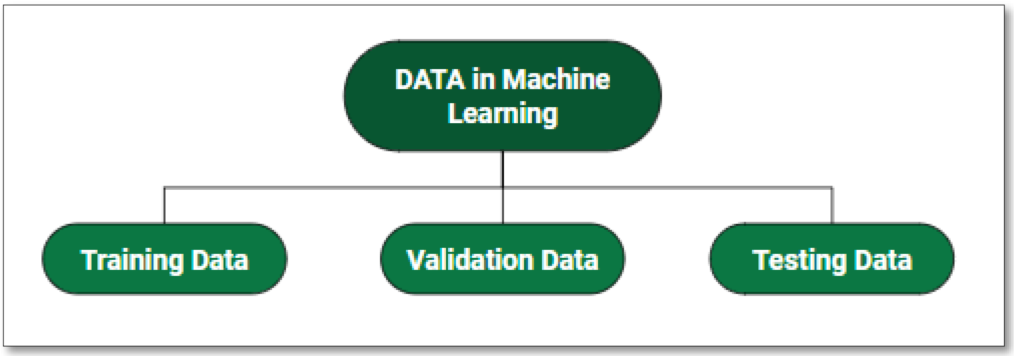


Figure 2.10 data splitting into 3 categories[49]

2.4.19 Properties of Data

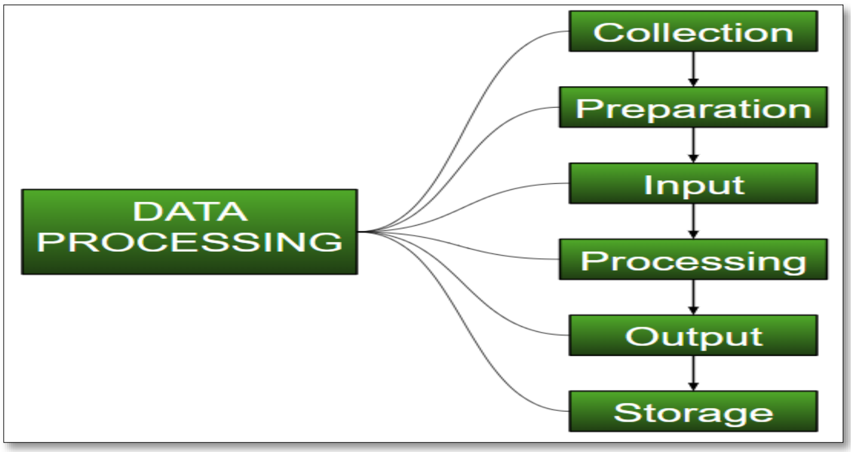
- Volume: (Scale of Data) With the growing world population and technology at exposure, huge data is being generated each and every millisecond.
- Variety: Different forms of data – healthcare, images, videos, audio clippings.
- Velocity: Rate of data streaming and generation.
- Value: Meaningfulness of data in terms of information that researchers can infer from it.
- **Veracity:** Working data certainty and accuracy.

The above-mentioned facts are just a glimpse of the actually existing huge data statistics. When talk in terms of real-world scenarios, the size of data currently presents and is getting generated each and every moment is beyond our mental horizons to imagine [51].

2.4.20 Understanding Data Processing

Data processing enriches data. Mathematical modeling, statistical distributions, and machine learning might automate this. This process may yield graphs, videos, charts, tables, pictures, and more. images, movies charts, and tables.

This seems easy, yet, Twitter and Facebook are huge administrative bodies like parliament and UNESCO, and health organizations must follow a very methodical and organized process Figure 2.11 shows the date processing [51].



.

Figure 2.11 data processing steps[51]

- **Collection**: Machine learning demands good data. Data.gov.in, Kaggle, and the university of California Irvine UCI dataset repository provide trustworthy data. Students use top study tools for tough exams. This improves learning.
- Costly and time-consuming data collecting: Organizations and researchers must select what data they need to work.

  - Example: The Facial Expression Recognizer needs many photographs of different human emotions to improve. Reliable data ensures model accuracy and dependability.[49]
- Preparation: The system may reject the data. This requires gathering data from numerous sources, interpreting it, and producing a new dataset for future analysis. Prepare manually or automatically. Numerical data compilation accelerates model learning.
  - Example: It is possible to turn a picture into a matrix of N-by-N dimension
  - , the value of each cell in the matrix will represent a pixel in the image.
- Input: Data may be unreadable. This data requires conversion. This needs precise calculation. MNIST Twitter, audio, and video snippets may give data.
- Processing: Algorithms and ML approaches must execute instructions across a huge amount of data with precision and optimum calculation.
- Output. In this level, the machine produces meaningful findings that users can understand. Reports, graphs, and videos are outputs.

Storage This is the final step in which the obtained output and the data model data and all the useful information are saved for future [48]

2.5 Image processing is defined

Digital image processing involves computer-based algorithmic image editing. Face recognition, object identification, and photo compression need preprocessing. Many other apps need it too. Image processing may enhance or reveal information. [50].

2.5.1 Image Processing Basics

Scientific observation has always been visual Orally and manually recording experiment outcomes was the sole option.

Photography documented findings. Astronomy, photogrammetry, and particle physics use photography scientifically. Astronomers measured star magnitudes and photogrammetry created terrain maps from aerial photos. Thousands of hydrogen bubble chamber photos identified numerous fundamental particles. Manual assessment took longer. Automated optomechanical devices exist. They were personalized. Quantitative image assessment wasn't popular then [52] .

2.5.2 Digital Image Fundamentals (DIF)

Computers process digital images. Digital image elements have locations and values. Image, pel, and pixel elements. Pixels comprise computer images. Images are two-dimensional functions that measure brightness and color from 3-D sceneries. Images are two-dimensional functions f(x, y) in plane coordinates. Amplitude is picture intensity at any place. Gray level describes monochrome visual intensity, Figure 2.12 shows digital fundamental system [53].

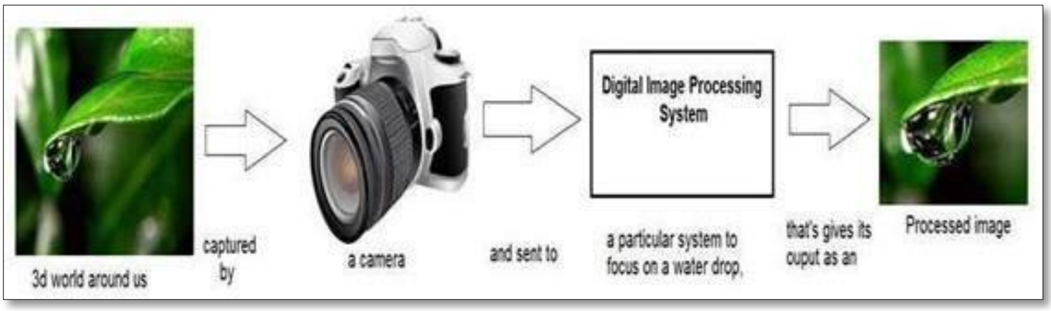

3d world around us
captured by
a camera
and sent to
Digital Image Processing System
a particular system to focus on a water drop,
that's gives its ouput as an
Processed image

Figure 2.12 Fundamentals of Digital Image Processing System

2-D photographs create color images. Red, Green, and Blue make up an RGB color picture. Color photographs may be processed using monochrome picture technologies [54].

2.5.3 Applications of Digital Image Processing

Since digital image processing (DIP) has very wide applications and almost all of the technical fields are impacted by DIP, A few key notes about DIP applications [55]

1) Digital image processing has a broad spectrum of applications, such as
    a) Remote sensing via satellites and another spacecraft
    b) Image transmission and storage for business applications
    c) Medical processing
    d) RADAR (Radio Detection and Ranging)
    e) SONAR (Sound Navigation and Ranging)
    f) Acoustic Image Processing (The study of underwater sound is known as Underwater Acoustics or Hydro Acoustics)
    g) Robotics and automated inspection of industrial parts
2) Medical applications:
    a) Processing of chest X-rays
    b) Cine angiograms
    c) Projection images of trans axial tomography and
    d) Medical images that occur in radiology nuclear magnetic resonance (NMR)
    e) Ultrasonic scanning

2.5.4 Essential Procedures for Processing Digital Images

There are two categories of the steps involved in the image processing:

1) Methods with images as inputs and outputs

2) Methods in which the attributes extracted from the images are returned as outputs.

a) picture Acquisition: A digital picture may be supplied. Image capture usually includes scaling.

b) Image Enhancement: One of the simplest and most appealing digital image editing professions. This brings to light previously unseen details or characteristics. Image editing is very individualistic.[54].

c) Image Restoration: Enhancement. Mathematical or probabilistic models restore pictures objectively. Human preferences determine enhancement. discussing "good" enhancement results [54].

d) Color Image Processing: Internet usage of digital photos has made it important. Models for colors and software for manipulating images are essential components of color image processing.

e) Morphological Processing: It extracts picture components for object form and boundary representation. Automated inspection uses it most.

Figure 2.13 shows the fundamental steps in digital system, Figure 2.14, and 2.15 Shows the image enhancement and image restoration.

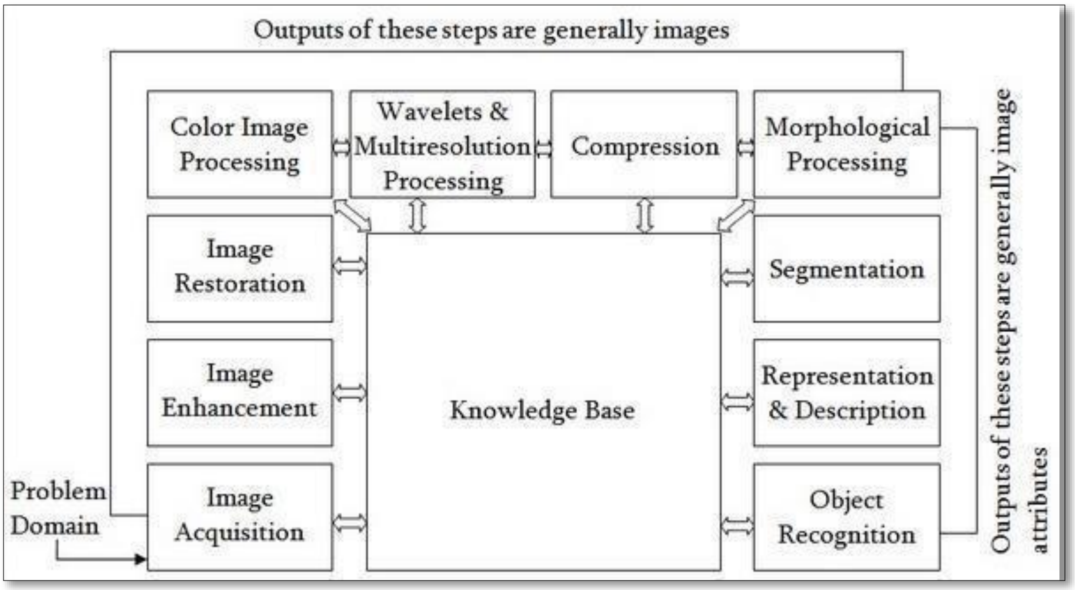


Figure 2.13 fundamental steps in digital image processing

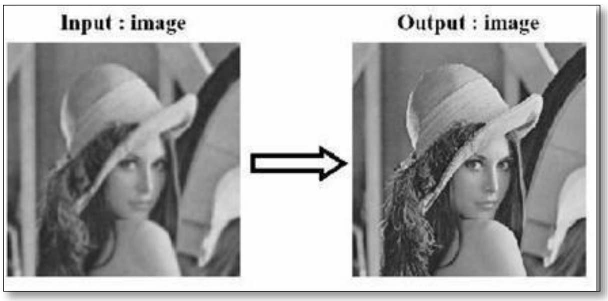


Figure 2.14 image enhancement

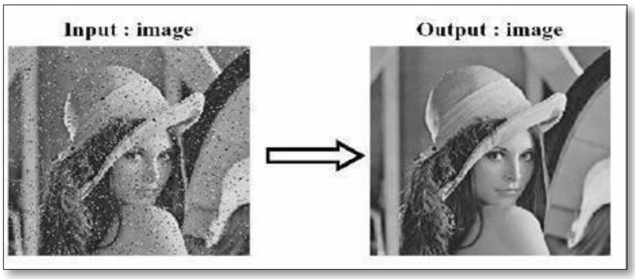


Figure 2.15 image restoration process

2.5.5 Representation and Description

It comes after the raw pixel data from the segmentation process, which determine the points that make up the image's borders or regions. In either scenario, the data must be converted before it can be processed by the computer. Recognition is the act of labeling a thing based on its characteristics. Image processing is finished by software that utilizes artificial intelligence

2.5.6 Sampling and Quantization

Digitize felt data before producing a digital picture. Sample and quantize. Image amplitude and x-y coordinates may be continuous Before digitizing sample coordinator ,

Image amplitude and x-y coordinates may be continuous. Before digitizing, sample coordinates and amplitudes. Amplitude is quantized picture spans AB line. Sample line AB at equal intervals to simplify this function [54].

2.5.7 Thresholding

Thresholding simplifies image segmentation. Thresholding converts grayscale to binary. Thresholding simplifies image segmentation.

Thresholding converts grayscale to binary. Thresholding classifies pixels as objects or backgrounds based on their brightness. Other forms include threshold below, the reverse of threshold above, threshold inside, where a pixel is deemed a "object" if its value falls between two thresholds, and threshold outside, the opposite of threshold inside [56].

1. Threshold Selection. The threshold value is the most important decision in thresholding. Customers or thresholding algorithms may choose a threshold. If object pixels are brighter than background and average, picking the mean or median is easy [54]. Brighter ones should be brighter.
2. Adaptive Thresholding. uses varying thresholds to picture regions. Local or dynamic thresholding is comparable. Organize things using the backdrop, foreground, or both. Gauss's entropy-based approaches use foreground, background, cross-entropy, etc. Object attribute-based methods compare gray-level and binarized images. It might be edge coincidence or hazy form likeness. Higher-order probability distributions and pixel correlations for spatial analysis. Local picture features affect pixel threshold values.[55]
3. Multiband Thresholding: Color photographs may threshold. One method applies a threshold to each (Red, Green, Blue) RGB component of the picture and then ANDs the results. One way. This illustrates how the camera and computer store data, but not how humans see color. (for hue, saturation, lightness) HSL and for (hue, saturation, value) HSV color models gain popularity [54].

2.5.8 Image Segmentation Computer Science

Digital imaging and computer vision need picture segmentation. This process breaks a digital picture into pixels. Image segmentation simplifies, clarifies, and simplifies [57]. Photo segmentation finds lines, curves, and other things. Pixels with the same label share properties in image segmentation.

Image segmentation generates contours or full-image segments. Image segmentation provides outlines that can be interpolated into 3D reconstructions using machine cubes [57].

2.5.9 Classes of Segmentation Techniques

There are two classes of segmentation techniques.

1) Classical computer vision approaches
2) AI based techniques

- ➢ Groups of image segmentation
  - a) Semantic segmentation classifies every pixel. When all figures' individuals and backgrounds are split as one object [58].
  - b) Instance segmentation assigns an object instance to each pixel. It recognizes each visual item. When each person in a figure is subdivided as an item [59].
  - c) Panoptic segmentation uses semantic and instance segmentation. Panoptic segmentation, like semantic segmentation, classifies each pixel. Panoptic
  - d) segmentation separates class instances, unlike semantic segmentation [60]

2.5.10 Convolution Neural Network (CNN)

CNNs (ConvNets) are used in deep learning. Examines images [40] Due to its shared-weight design of convolution kernels or filters that slide along input features and create translation-equivariant feature maps, CNNs are sometimes called shift invariant or space invariant artificial neural networks (SIANN). Shift- or space-invariant artificial neural networks are CNNs. Most convolutional neural networks are not translation-invariant owing to input down sampling [61]. Image and video recognition, recommender systems, image classification, segmentation, medical image analysis, natural language processing, brain– computer interfaces, and financial time series employ them. Medical image analysis, segmentation, classification, natural language processing [62]

2.5.11 Foundation of Convolutional Neural Network

CNN Concepts Neurophysiologists David Hubel and Torstein Wiesel published "Receptive Felds of single neurons in cat's striate cortex" in 1959 [63]. Cat neurons were layered in this study. These layers combine local features to discern visual patterns. This became Deep Learning's foundation.

Categorizing visual patterns. Figure 2.16 depicts the original theoretical CNN architecture. CNN framework [64]. This technique outperformed Noncognition in identifying MNIST handwritten digits[65]. LeNet-5[65] recognized visual patterns from raw input images without feature engineering using error backpropagation procedure. CNN struggled in tough conditions once LeNet-5 was discovered. This was owing to a lack of training data, algorithmic innovation, and computing power, After AlexNet's success, several CNN models were developed and used to other computer vision and natural language processing subfields [61] .

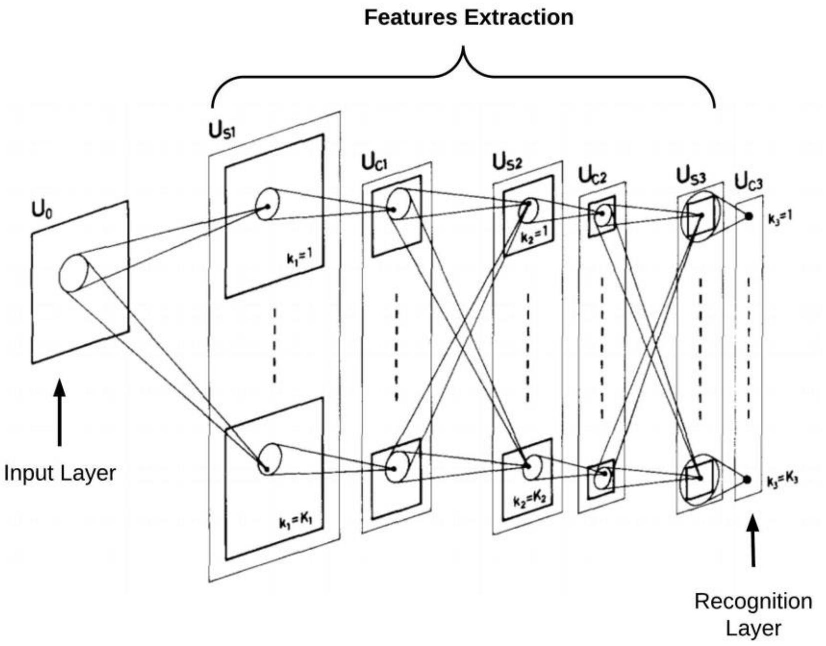


Figure 2.16 Schematic diagram illustrating the interconnections between layers in the recognition, Kunihiko Fukushima [66]

2.5.12 Concepts of Convolutional Neural Network

FC-layer networks struggle to detect more abstracted things like geographical data. The CNN, also known as ConvNet, generalizes better than FC-layer networks because to its deep feed-forward design. A deep CNN model with a few processing layers may learn many picture features at various abstraction levels. Initiatory layers learn high-level traits, whereas deeper layers acquire low-level properties. Figure 2.17 depicted the fundamental conceptual paradigm that CNN uses, and the remaining parts of this article discussed the many sorts of layers.

In the field of computer vision, why do convolutional neural networks have greater weight than traditional neural networks:

1) CNN's weight-sharing mechanism minimizes trainable parameters. avoid overfitting and enhance generalization.
2) CNN's classification and feature extraction layers learn continuously, organizing and using obtained features.

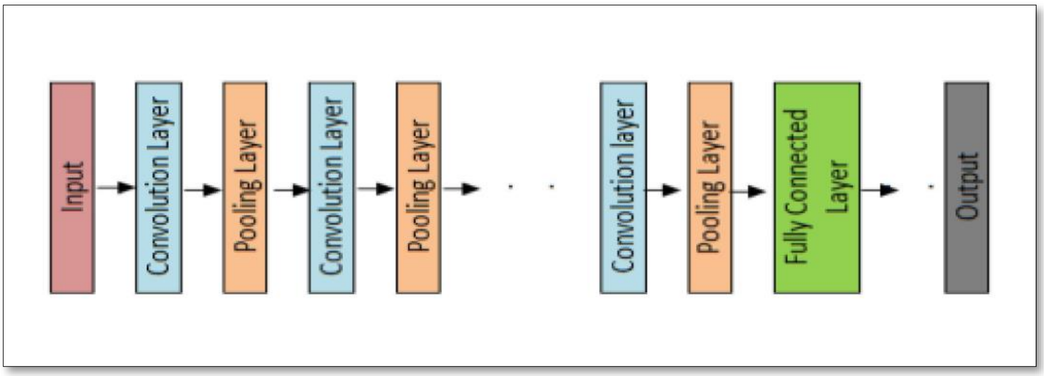


Figure 2.17 Conceptual model of CNN [67]

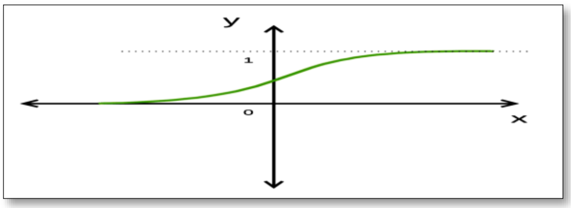
3) Non-convolutional neural networks complicate massive network implementation. CNN is presently utilized for image classification, object identification, face detection, voice recognition, vehicle recognition, facial expression recognition, text recognition, and other computer vision applications.

2.5.13 Activation Functions

Neural network activation functions must connect input to output. The input value is the weighted sum of each neuron's 10 inputs plus bias. The activation function regulates whether a neuron activates and outputs in response to an input. CNN designs employ non-linear activation layers after each learnable layer (weighted convolutional and FC layers). For accuracy. CNN models transform inputs to outputs and learn more complicated things due to layer non-linearity.[67]

- Sigmoid function. The sigmoid activation function outputs [0,1] for real values. S-shaped sigmoid curve. Figure 2.18 (a) represent the sigmoid mathematical forum and the Figure 2.18 (b) output shape.

$$f(x)_{sigm} = \frac{1}{1+e^{-x}}$$



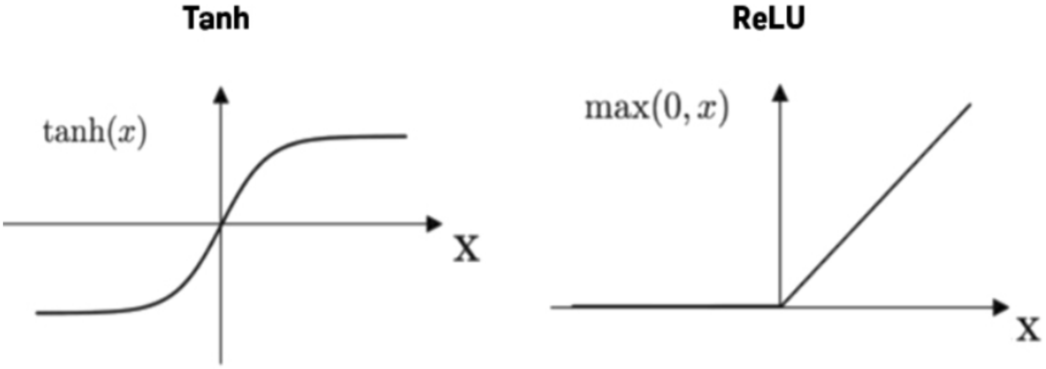
(a) (b)

Figure 2.18 Sigmoid function [67]

- **Tanh function**. The Tanh activation function is used to bind the input values (real numbers) within the range of [-1, 1]. The mathematical representation of Tanh is show In Figure 2.19.a
- **ReLU Function**. Most convolutional neural networks employ the Reciters Linear Unit (ReLU) . It makes all entered values positive. ReLU is a low-compute algorithm. Mathematically, ReLU is:

f(x)ReLU=max (0; x). it represents in Figure 2.19(b)

ReLU activation may create serious complications. Erroneous wfunction may update the weights such the neuron never activates again. [67]



$f(x)_{tanch} = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$ `f(x)=max(0,x)`

(a) (b)

Figure 2.19 (a) tanh function, (b) Relu function [67]

2.5.14 Normalization.

In this step, normalize the dimension of the data sample, which includes both the train dataset and the validation and test datasets, by dividing each dimension by its respective standard deviation, as illustrated in Figure 2.20. In mathematical terms, the procedure would be carried out as follows

$$:x = \frac{x}{\sqrt{\frac{\sum_{I=1}^{N}(X_I - X^*)^2}{N-1}}} X^1\text{=X-}X^*\text{,And, } X^*\text{=}\frac{1}{N}\sum_{I=1}^{N} X_I$$

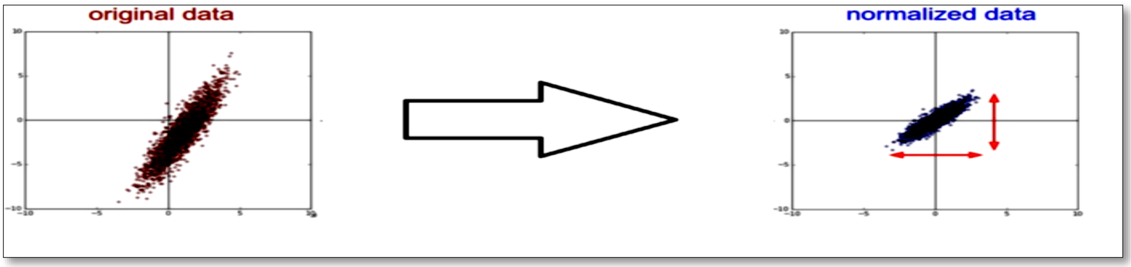


Figure 2.19 Normalization of data[67]

Chapter Three

Proposed Methods

Chapter Three

Proposed Methods

3.1 Introduction

The vector extends in that particular direction. Based on our investigation, there are many alternative routes to reach the target. The goal was to proactively avoid a stroke, allowing for enough preparation to navigate through it safely.

Stroke is a major global health problem that affects people of all backgrounds and ages. Stroke is an actual leading cause of mortality worldwide. and it endangers the lives of those who encounter it. Our primary goal is to develop a reliable early warning system that can anticipate strokes, and we'll be using scientific principles to do so. The initial stage of the procedure is focused on collecting information from the person. There are a few ways to go about it.

1) Collect data by CT (computerized tomography)
2) Collect data from MRI (magnetic resonance image)
3) Collect data from a pet- scan (positron emission tomography)
4) Collect data from an Electroencephalograph –(EEG) device.

To get the data, two separate methods were used under supervision. in this case, two independent channels have been established since two separate methods were used to achieve the goal: first, data was collected by an electroencephalogram (EEG), and subsequently by a magnetic resonance imaging (MRI) and computed tomography (CT) equipment. It all starts with using an electroencephalogram (EEG) instrument to gather data, while the second method uses an MRI, CT device. As a result, offering alternative approaches to data modification is crucial. Moreover, need to make use of certain tools and methods. To get started, used the EEG equipment to collect the data and ran it via MNE-PYTHON [68]

The stroke is divided into two types.

1) **ischemic stroke, or**
2) **a bleed (hemorrhagic stroke).**

In order to collect the data, two separate methods were used, following the instructions of the supervisor. Initially, EEG data was collected, and then, MRI data was collected; hence, in this case, two different channels were installed as two independent methods were used to achieve the goal. In the first method, an electroencephalogram (EEG) device is used to collect data, whereas in the second, a magnetic resonance imaging (MRI) method is used. Two different methods were used to gather data: one that used an EEG device and the other that relied on an MRI equipment. This means you'll need to employ a wide range of programmes and methods, and you'll also need to provide other approaches to data manipulation. First, we used the EEG equipment to collect data, and then we employed MNE-PYTHON to accomplish our purpose [69].

3.2 General Work Headline

The block diagram below shown the proposed methodology delineates two pathways for stroke identification in Figure 3.1: the image processing pathway, which could be exemplified by MRI and CT scans, and the electroencephalogram route.

Collecting data from participants after the date, extracting characteristics, and selecting relevant traits are the main focuses of the two methodologies. The MNE-PYJTHON software does a good job of processing EEG data and extracting characteristics.

Using image processing methods, a Python programmer was able to alter the signals after features were extracted from the IP path signals. Before being sent to machine learning, all features from both pathways were organized into a csv file. Stroke detection methods abound.

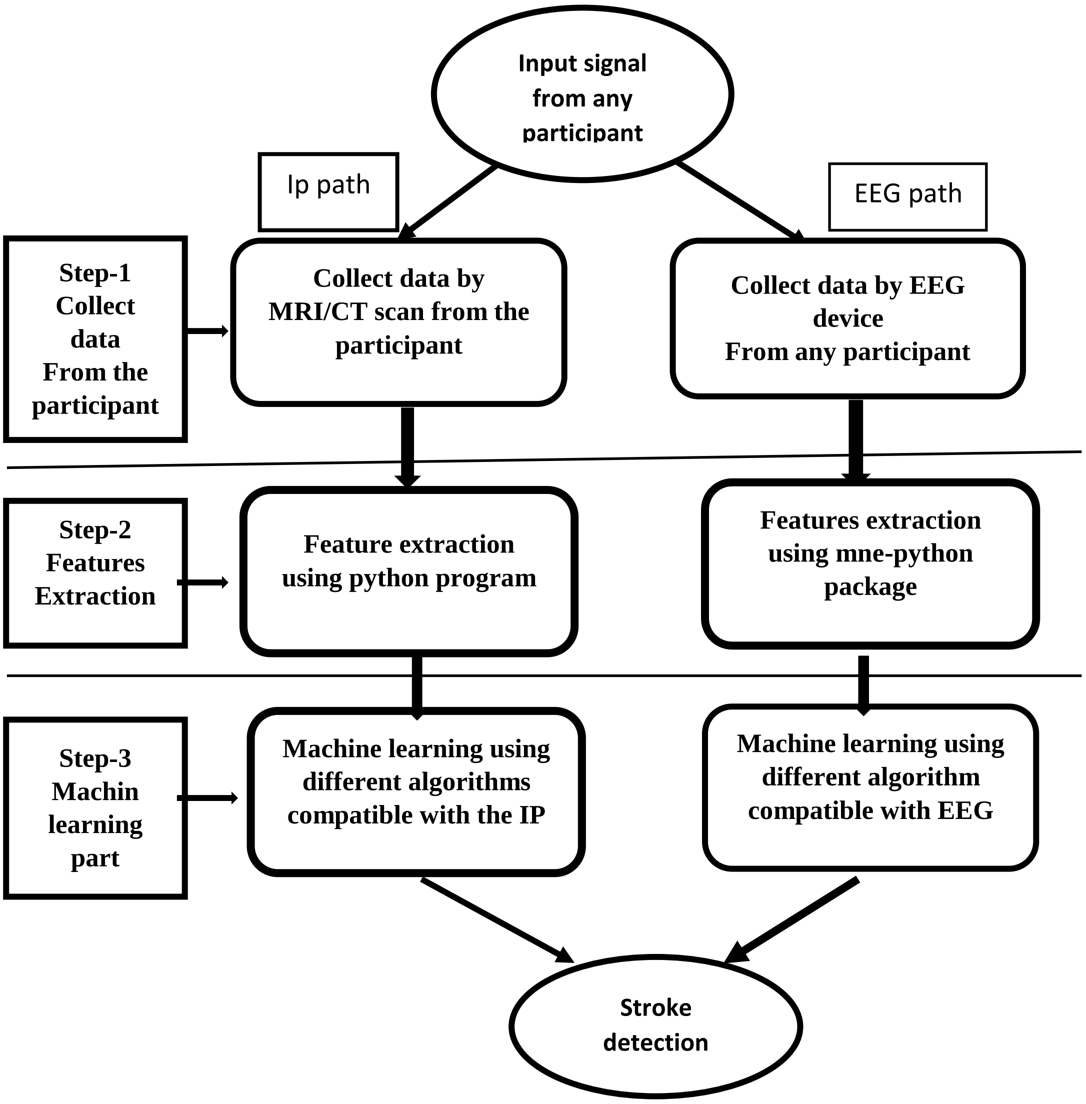

Input signal from any participant
Ip path
EEG path
Step-1 Collect data From the participant
Collect data by MRI/CT scan from the participant
Collect data by EEG device From any participant
Step-2 Features Extraction
Feature extraction using python program
Features extraction using mne-python package
Step-3 Machin learning part
Machine learning using different algorithms compatible with the IP
Machine learning using different algorithm compatible with EEG
Stroke detection

Figure 3.1 the main idea behind the suggested way to do the work.

The stages that may be listed and explained in further detail are evident in the preceding Figure 3.1. *Data collection* is the first phase and may be done in two ways. The image data collection from MRI or CT scan devices (IP) and the EEG collection by EEG devices (representing time series)

The second: stage entails implementing the suggested approach for bidirectional feature extraction.

Third Step: depending on the kind of job, such as detection or prediction, employ a machine learning approach with two possible paths and choose an algorithm that works well with time series (EEG) or image processing (IP).

3.3 Several factors point in that direction

Variations in cerebral blood flow in blood vessels may identify stroke; the EEG has been found to be helpful in detecting a range of other brain-related ailments, such as rapid eye movement (REM), sleep, and aberrant brain waves. This direction focuses on "ischemia stork prediction," using electroencephalography (EEG) as the primary direction and all data stored there.

Six locations on the head were used to record the electroencephalogram, representing six distinct signals. Two unused neurons, plus two frontal lobe neurons, and two temporal lobe neurons (one on each side of the head, next to the ear) all together make up a reference. Stroke can be diagnosed suffice by observing a change in blood pressure between the left and right hemispheres of the brain. The six signals are sent to the skull at a total of six separate locations. Features are extracted to identify this alteration (in a 30-second period) as a result of ischemic stroke, which may be checked by measuring the amplitude and frequency of the signals. In this situation, measuring the amplitude of the signals across time will reveal any time-frequency changes in the signals. In order to get the feature out of the signals, as part of the proposed technique, choose your top two choices from the outset. In the proposed select-two method, the signals are initially used to gather features. To extract features from signal data, scientists use the proposed choose one of two options

1) Frequency domain
2) Time domain

There are two groups of characteristics that relate to the frequency domain and another set of features that correspond to the time domain, these features get to signal that a stroke may occur at any moment, think of it as an early warning.

These properties have been used together to train multi types of algorithms like

1) MLP,
2) Bootstrap (Extra-Tree and)
3) Decision-Tree model.
4) Random forest model

With an accuracy of 95% and an area under the ROC curve of 0.85% on test data, this strategy has been shown to be effective. These results showed promise and might be used to the early diagnosis of Transient Ischemic Attacks (TIAs).

Due to the non-stationary and nonlinear nature of EEG signals, most feature extraction approaches are based on a single domain; yet, that study makes two significant advances to the identification of ischemic stroke. The initial contribution made use of EEG signals to identify strokes using multi-domain characteristics.

Information extraction from a single domain is a complex operation. First, we've made progress towards a more equitable training set, and second, we've found that extracting features from both the frequency domain (PSD, FFT, ratio) and the time domain (one parameter per channel, utilize 4 channels) yields the most accurate results. making use of Bootstrap models. And get an accuracy of 95% or better has a more universally beneficial roc curve. To improve the overall area under the Roc (receiver operation curve), by reducing the number of false negatives (FN), and to achieve an accuracy of 95%, imbalanced training is commonly used in the medical field because the number of people affected by a particular case is typically a small percentage of the general population.

3.4 The dataset description

Table 3.1 Dataset, consists of 5 groups that represent the EEG signal specification types. And do elaborate on each point.

1) File name. File names denote EEG file categories.
2) NO. of channel. represent the channel numbers and it refer to the device manufacturing.
3) Time recording. it represents the time required to record signals and it depended on the person who supervises it.
4) Sampling frequency. represent the frequency that used to record the data, it.
5) depends on the person supervising the recording.
6) Type of file. It indicates file format and always relies on Device manufacturing facility.
7) Number of participants.. The total number of people involved may be seen here. The resource displays and downloads all file formats [69][70]

Pick up a hundred perceptron’s from the table. To begin, the process has to gather a massive quantity of data in order to provide useful results. If you want to move in the path of machine learning, satisfying the massive quantity of data is crucial. The table shows a variable sample frequency; therefore, this must be taken into account for any analysis.

Table 3.1 dataset details

File name	No. of channel	Time recording	Sampling frequency	Type of format extension	No. of participants
Ds00291	32 Ch	1023sec	250 Hz	Set file format	20
Dsexg15	21 Ch	182 sec	300Hz	EDF file format	36
Ds3194	19 Ch	600 sec	200 Hz	EDF file format	14
Ds3195	19 Ch	774 sec	250Hz	Set file format	10
Ds2778	40 Ch.	192 sec	512 Hz	BDF file format	20

3.5 The stroke-related EEG backdrop

Electroencephalogram (EEG) readings It's a great way to gather more precise data about the brain, and it can tell you whether or not a certain signal has a typical pattern of behavior. The signals' shapes provide a first indication of their temporal and frequency regularity. Specifically, knowing that there is a problem with the way a person's brain functions is crucial.

The scalp is used to pick up the vast majority of the cerebral signal, and the range of frequencies for all signals is between 1 and 20 hertz. Scalp EEGs can detect frequencies between 1 and 20 hertz. Most clinical applications of EEG make use of waveforms, which are classified according to frequency range as delta (δ), theta (θ), alpha (α), and beta (β) (Figure 3.2). The actual assault Changes in cerebral blood flow (CBF) are a common cause of this disorder, and a change in the form pattern of an EEG single may be used to diagnose it [71].

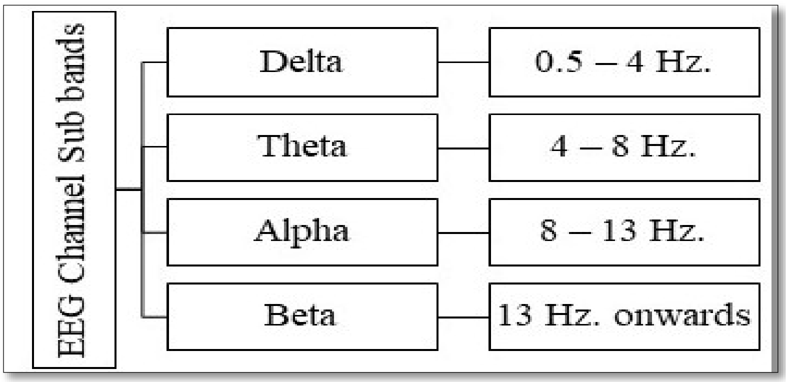


Figure 3.2 EEG Channel Frequency bands range

As a result of this change, the delta band, which operates at very low frequencies, has shrunk, joining the beta and theta bands that function at higher frequencies [72]. When one hemisphere has a stroke, the other hemisphere's power density ratio between the band’s shifts [73]. Using a more targeted MRI scan might increase the accuracy of the diagnosis. besides the patient's previous medical records. As features for ischemic stroke detection, in order to accurately address the stroke situation, the power spectrum density (PSD) ratio between EEG bands and channels may be examined and computed to act as a warning. Similarly, the power ratio between bands and change between channels can serve as indications to stroke detection[74], are just a few examples of classifiers that incorporate PSD in their training. With these settings, the KNN classifier achieved an accuracy of 85%. with 85% precision using ANN [74] and 85%-time savings.

When the frequency domain is enhanced, the RPR reaches 93%. This is yet another measurable and quantifiable quality. Moreover. Spectrograms are used as features in the training of a convolutional neural network (CNN). When an ischemic stroke is detected, the EEG signal is altered, and those alterations are mapped using a convolutional neural network [75]. A spectrogram is a graphical representation of the relationship between signal amplitude and frequency.

The EEG signal is broken up into 30-second intervals that overlap by 20% of the total recording duration. When training a convolutional neural network, its input is the spectrogram for each time interval. Obtaining the capacity to view and alter the properties, permitting further processing of EEG data, may be done in a number of different ways [76].

The power spectrum density is a feature that was chosen to be calculated for four channels using the welch method. This method has the advantage of robustness, which ensures that there is no invalid frequency in the density analysis. On the other hand, this method's disadvantage is its windowing, which leads to distortion of the resulting density**.**

Over the course of this job. used a combination of the relative power ratio (RPR) across the hemispheres and channels, as well as detrended fluctuation analysis, to train four models (an MLP, a DT, and an ET). The RF algorithm, or random forest, Because of the nature of the data we're dealing with—imbalances—we use Classifiers such decision trees, boosted trees, and random forests. When it comes to handling unbalanced datasets, these classifiers excel above their more traditional counterparts. [77].

This study includes healthy and stroke subjects. The stroke case has fewer participants than the healthy case, hence the statistics are skewed. In imbalanced data, the target class has an unequal distribution of observations. This suggests one class label has many observations, whereas the other has few. This example will help us understand imbalanced dataset handling. [78]

3.6 Strategy and Method

The methodology proposed is based on supervised learning.as show In The methodology proposed is based on supervised learning.as show In Figure 3.3 and 3.4 the EEG signals is analysis with two domains

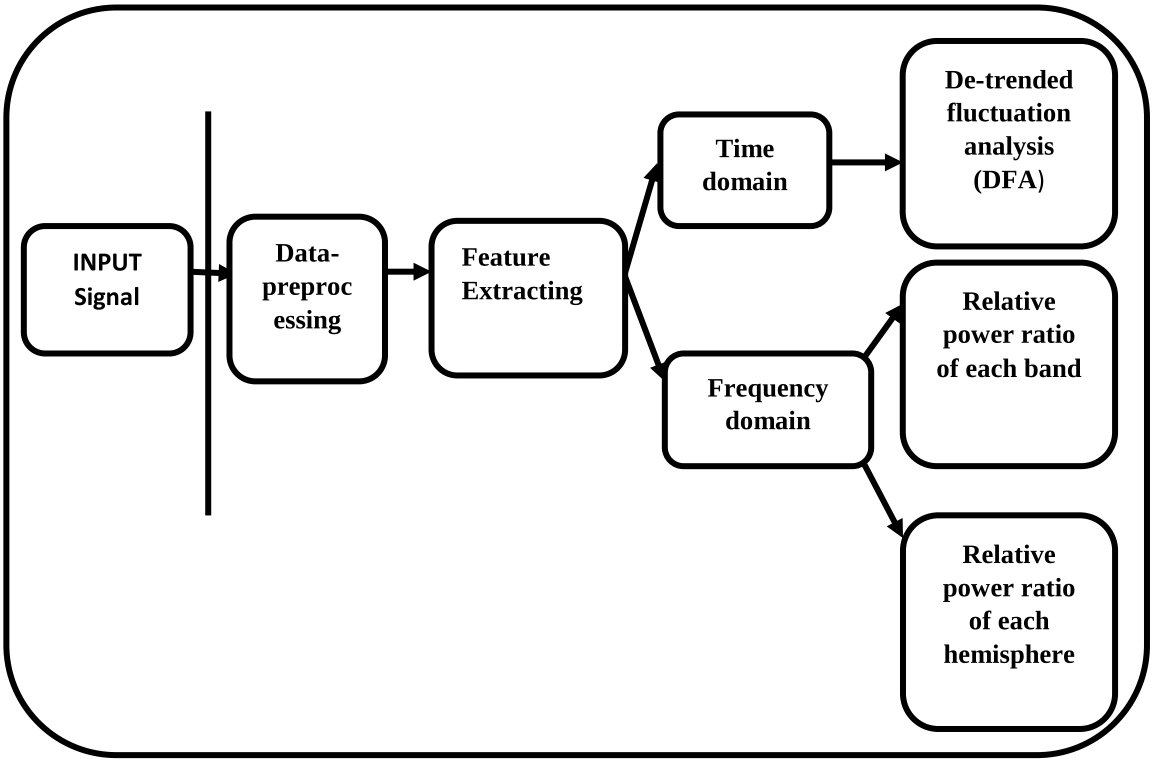

INPUT Signal
Data-preprocessing
Feature Extracting
Time domain
Frequency domain
De-trended fluctuation analysis (DFA)
Relative power ratio of each band
Relative power ratio of each hemisphere

Figure 3.3 First Stage Data Cleaning and Feature Extractions

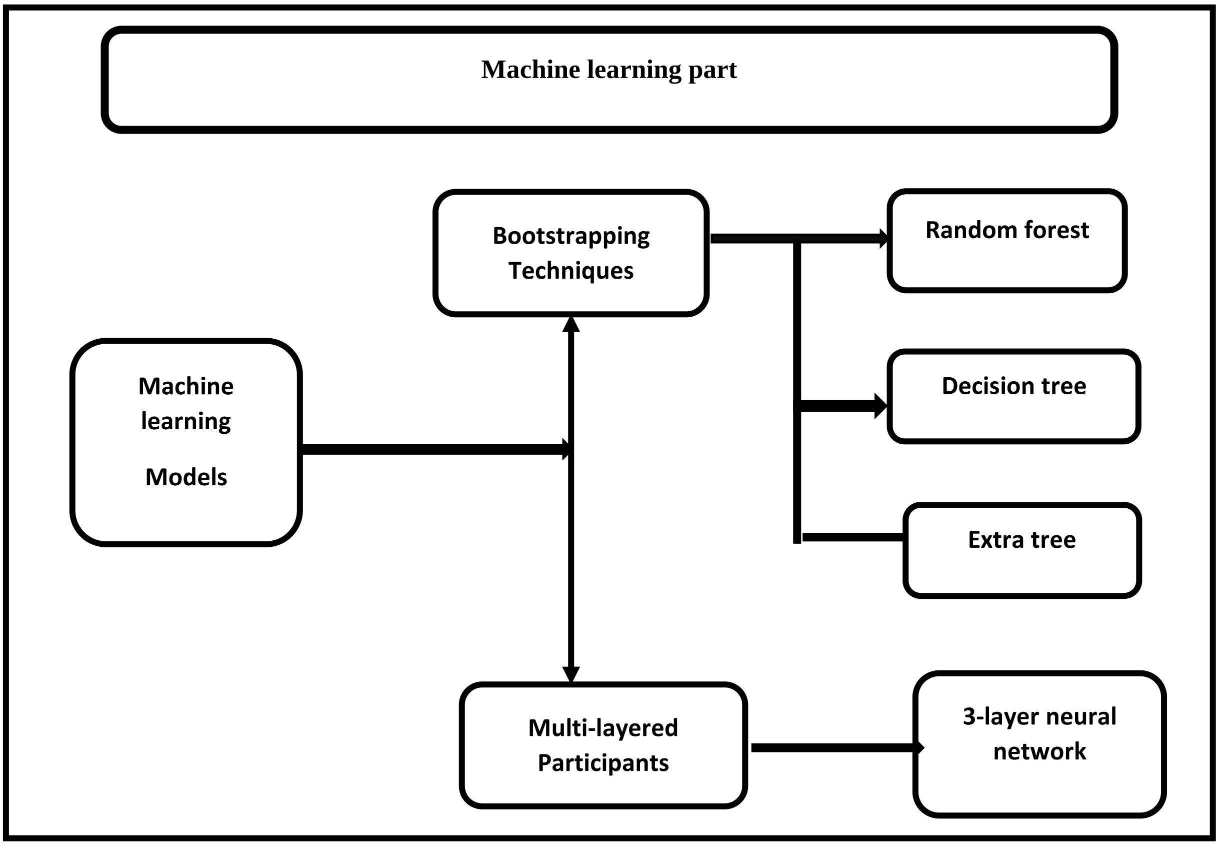


Figure 3.4 Second Stage training and testing of machine learning models

No study that has been done in the past has specifically leveraged multi-domain characteristics to identify ischemic stroke using EEG data. The procedure shown in (Figures 3.2,3.3) which includes several phases, primarily consists of two parts: the first portion collects the data (EEG-based), followed by cleaning the data and feature extraction.

- First stage. collect data+ feature extraction Figure 3.3, In that work, collect data from one hundred participants, and each participants receives 25 attributes that represent all features within two domains (time and frequency). As a result, the new dataset contains one hundred times twenty-five representations of all features, and this is the first stage of operation.
- Second stage. second stage Figure 3.4, and more explain In Figure 3.5 models of machine learning, including training and testing. This point can be started after step one has been completed; at this point, many types of multiple classification algorithms and prediction models used ; this point can be started after step one has been completed; can selected multiple classification algorithms; the type of selection depends on whether the primary goal is classification or prediction; in that work, selected supervised classification and prediction as the way to go as the heading for that work.

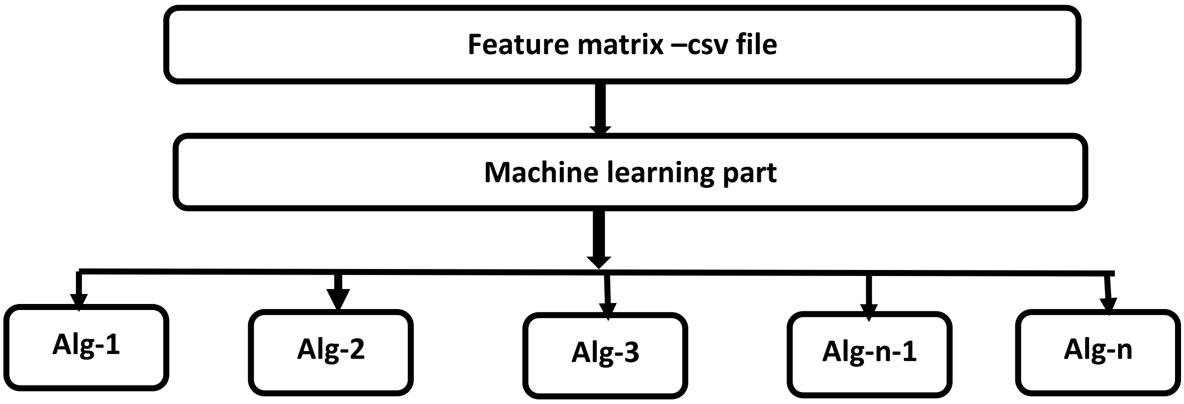


Figure 3.5 machine learning part procedure

In the block diagram shown above (Figure 3.5), the (Alg) refer to algorithm, many different kinds of algorithms exist, and can be choose one appropriate for the task. The focus of the work is on developing methods for classifying and detecting.

3.7 Data Pre-Processing

Once I had obtained the EEG signals, I imported them into the job from the reference [69][70]. For further information, please refer to table 3.1.

In the pre-processing phase of the data, there are three fundamental components that make up the phase.

referencing data*, *segmenting*, and *cleaning up artefacts.

3.7.1 Data referencing details

- The EEG signals collected from six electrode point from the head show in Figure (3.6), C1(F_{P1}), C2(F_{P2}),C3(O1),C4(O2), and 2 electrodes represent the reference ,the electrode C1,C2 represent the front side of the head, the electrode o1,o2 represent the back side of the head
- The reference definition with EEG

The reference electrode that is used to record EEG data is often called the "common" reference for the data. When every channel uses the same reference, this is what happens. While EEG recordings are being made, the red electrode in the picture below is TP10, which is a mastoid. Other recording references include linked mastoids (usually digitally linked mastoids), the vertex electrode (Cz), single or linked earlobes, or the tip of the nose.

Data may be recorded without the need for a reference in certain systems, such as BIOSEMI Active Two, which include active electrodes. In this scenario, a reference be is required to be selected ad hoc as the data is being imported. If not, the data may include 40 dB of unwanted noise [76].

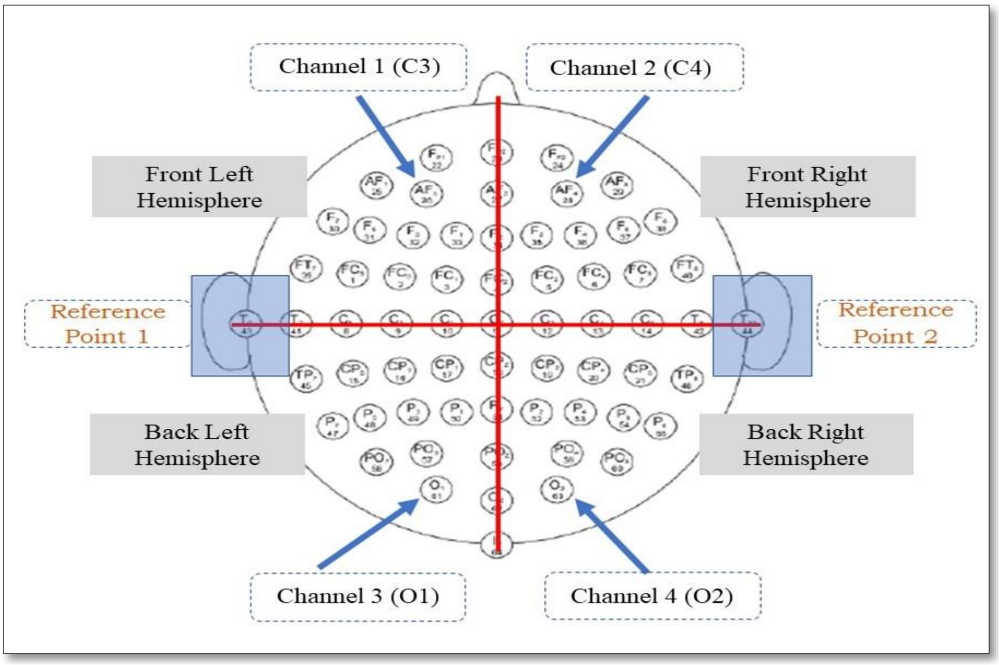


Figure 3.6 Blue arrows indicate the frontal and parietal lobe EEG channels C3, C4, O1, and O2.[22]

$$x_c(n) = r_c(n) - \frac{r_{ref1}(n)+r_{ref2}(n)}{2} \quad [22] \qquad \textbf{(3.1)}$$

where xc(n) is the referenced EEG signal for channel C3, C4, O1,O2, rc(n) is the raw EEG signal for channel C3,C4,O1,O2, and rref1 (n) and rref2 (n) are the raw EEG signals for the two reference channels.

3.7.2 Grouping by Type

To better understand the characteristics of a given time series, it may be broken down into smaller, more manageable chunks using a technique called time-series segmentation. The goal of doing so is to expose the underlying characteristics of the time series. The goal of this approach is to extract as much useful information as possible from the time series. Speaker divarication is only one use of time-series segmentation.

In most cases, the EEG waves will not be completely still. In this studies, I just need to use four electrodes on each participant's scalp to record their electrical brain activity (EEG). Segmented EEG data of 30 seconds is often used to indicate the existence of a broad, stationary signal. This is shown when the data are arranged chronologically. Data analysis requires 30-second time segments, called epochs. Each channel's signal is divided into 7680 samples over 30 seconds after data is referenced, and the fs frequency is set at 256 hertz. [79]

3.7.3 Artifact Rejection

The electroencephalogram (EEG) is vital for brain activity and behavior analysis. Unfortunately, artifacts in recorded electrical activity will always interfere with EEG signal interpretation. Thus, systems that can recognize and recover clean EEG data during encephalogram recordings are essential.

Many artefact removal technologies exist, but research continues. This study examines artefact cleaning procedures. Conversation begins with EEG data properties and artefacts. An overview and in-depth analysis of the newest and most cutting-edge methods follows. In conclusion, comparative research helps customers choose application-specific solutions. [77]

- Different Categories of Artefacts

While gathering EEG data from recording equipment, signal artefacts become more prominent [80]. The integrity of EEG data may be compromised by these artefacts.

- ocular Artifacts

Significant artefacts are produced in the EEG recordings due to ocular artefacts [79]. Ocular artefacts originate in the eyes and spread over the scalp to be recorded as electrical brain activity (EEG). More precisely, corneal dipole orientation variations induce artefacts during eye movement, while corneal contact changes during blinking produce artefacts through ocular conductance [80].

- Artefacts of Muscle

It is a well-known and challenging issue because various muscle groups' activity may contaminate EEG readings. Any muscular contraction or stretch near the signal recording locations, as well as the patient talking, sniffing, or swallowing, might induce these artefacts [80]

- Heart artefacts

Cardiac artefacts may be created when electrodes are placed on or near a blood vessel [81], which moves owing to the heart's expansion and contraction. Such aberrations, known as pulse artefacts, with a frequency of roughly 1.2 Hz, may exist inside the EEG as a comparable waveform and are therefore difficult to eliminate [80]

- Single Artifacts Removal Techniques

Clinical artefact rejection often entails visually identifying and then manually removing the artefact signal. Several other techniques, methods, such as linear decomposition and reconstruction, time-frequency regression, and many more have been proposed for removing artefacts from EEG data.

- Inferential Techniques.

Conventionally, regression approaches have been used to clean up EEG data from unwanted noise. It is used on the premise that pure EEG data plus some amount of artefact make up each channel individually .

- Principal Component Analysis PCA, which uses covariance matrix Eigen values, is one of the simplest and most used BSS methods. This approach initially orthogonally transforms correlated variables into uncorrelated ones. Principal components (PCs) are uncorrelated variables [80]
- Component Analysis, Isolated. ICA is another approach that may deconstruct an observed signal into independent components (ICs) on the assumption that

the signal sources are instantaneous linear combinations of mental and artistic elements.

3.8 Feature Extraction

The feature must be select to help to satisfy the target, the target to discrimination the stroke and fix this case exactly so all features must be to be focused into that direction, in that work all features lies in two domain.

3.8.1 Time Domain Features

Long-term memory is measured statistically by the Hurst exponent. throughout the course of several years. It has to do with the time series' autocorrelations and the pace at which they decay with increasing time lag. As the time gap between the two values grows, this rate falls at a predetermined pace. The current state of affairs has been noted for some time, , the "Hurst exponent" and "Hurst coefficient" designations [82]. His name is also associated with the common coefficient notation H. representing the fractal dimension, is inextricably linked to the randomness measure H, which quantifies how "mild" or "wild" the unpredictability of a data series is [83]

A statistical metric sometimes known as the "index of dependency" or the "index of long-range dependence," the Hurst exponent has many other names. This is achieved by providing a numerical value for the proportional probability that a time series will substantially regress to the mean or cluster in a certain direction. High values in a time series are likely to be followed by additional high values and by similarly probable high values in the far future if H is between 0.5 and 1, a condition known as long-term positive autocorrelation. A score between 0 and 0.5 indicates that the time series is exhibiting long-term flipping between high and low values in nearby pairs. If this value is high, it means that the next value will likely be low, and so on, and so forth, for a very long period into the future. In actuality, however, H = 0.5 is appropriate for series whose autocorrelations at small delays may be either positive or negative, but where autocorrelation absolute values drop exponentially quickly to zero, implying that the series is in reality uncorrelated. For series where autocorrelations at small delays might be positive or negative, this is the critical value. Contrast this with the typical decline according to the power law, which happens between 0.5 and 1, and you get a picture of intrinsic persistence for H and 0-0.5 [82].

In that work used a ***python program***, the function can be implemented ,Python program (***hurst.py)*** attachment in Appendix A to compute ***Hurst exponent*** [83],

Mean value, variance, and standard deviation may also be computed for each channel. Here, the aforementioned computations stand in for the time domain characteristics.

3.8.2 Frequency domain features

Ischemic strokes, according to the medical literature, have an effect on the signals in the low-frequency bands; specifically, the delta, theta, and alpha bands [8]. There will be changes in relative power across channels since an ischemic stroke typically affects only one side of the brain [84]. Therefore, when an ischemic stroke occurs, variations in the power of each channel sub-band and the relative power ratio between channels serve as reliable markers.

- Measurement of Power in Sub-Bands

Welch's weighted overlapped segment averaging method estimates power spectral density by slicing the recorded signal into overlapped windows of length L, calculating modified periodograms of these windows, and averaging these modified periodograms; this yields the sub-band of the EEG signals for 4 signals (c3, c4, o1, o2). [85].

Welch method: Spectral density estimate is done using Peter D. Welch's technique. Physics, engineering, and applied mathematics utilise it to estimate signal strength at various frequencies. The approach uses periodogram spectrum estimations, which are obtained by translating a signal from time to frequency. Compared to the traditional periodogram spectrum estimating approach and Bartlett's method, Welch's method minimises noise in estimated power spectra but lowers frequency resolution. Because defective and finite data produce noise, Welch's approach is commonly wanted for noise reduction. The modified itch window periodogram is.

$$\bar{p}^{i}(f) = \frac{1}{LU} | \sum_{n=0}^{L-1} x_i(n)\omega(n)e^{-j2\pi fn} |^2 \qquad (3.2)$$

where U is the normalization factor for the power in the window function

$$U = \frac{1}{L}\sum_{n=0}^{L-1} \omega^2(n) \qquad (3.3)$$

Additionally, the window function is denoted by w(n). As a mean of these adjusted periodograms, we get the Welch Power spectrum.

$$\bar{p}(f) = \frac{1}{k}\sum_{i=0}^{k-1} p^{-i}\,(f) \qquad (3.4)$$

The study used a 50% overlapped sliding window with a 2-second window length (L = 2sec × fs = 512) to measure Power Spectral Density. The number of windows per epoch is T −1 = 29. Calculate the average power of sub-band b (δ, α, θ) for channel c (C3, C4, O1, O2) by

$$\bar{p}_{d}^{c} = \frac{\sum f_{min} \le f < f_{max} \bar{p}(f)}{n_b} \qquad (3.5)$$

In Figure 3.2, f_{min} and f_{max} represent the lower and higher frequencies of each sub-band, while nb = (f_{max} −f_{min})/fr shows the number of frequency samples for each range. For example, $f_r = f_s/L$, where fs is the sampling frequency.

The job's Python tool allows you to develop a Python programmed: [18] Welch technique power spectrum calculation.

Figure 3.7 shows the spectrum of the welch after the program execution that attachment **In Appendix A**, **welch.py** as a source program can be executed by python ver.3.

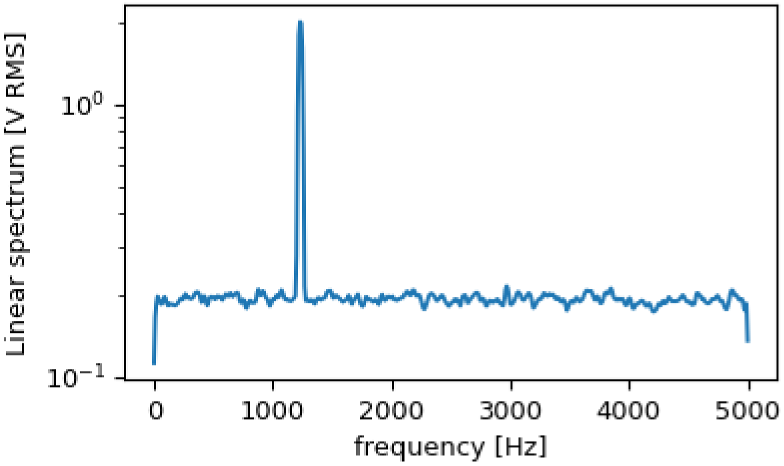


Figure 3.7 the output spectrum to the welch method to the welch.py program

There are two main point *calculate the relative power –sub band density.*

- Relative power sub band: It mean that the ratio between the one band of the frequency to another of summation to all bands, in that work, used 4 channels c ∈ {C3, C4, O1, O2}, The sub bands b ∈ {δ, θ, α},, so each channel contain 3 band in this case get 12 feature for each participant.

The relative power for each sub-band is calculated by (6):

$$\bar{P}_b^c \ / \sum_{c \in C} \bar{p}_b^c \tag{3.6}$$

- Relative Hemisphere Power**:**

The power density differential between the right and left sides of the head (or the front and rear) may be represented by the hemisphere shown in that section. Channels C3 and C4 are found in the frontal left and right hemispheres, whereas channels O1 and O2 are found in the posterior left and right hemispheres**.**

RPR(b)f h, that simple represent --- relative power ratio bands b ∈ {δ, θ, α}, f --- front, h--- Hemisphere the difference between **C3 and C4** for each sub-band in (3.7)

$$(|\bar{p}_b^{c3} - \bar{p}_b^{c4}|) \, / \, (|\bar{p}_b^{c3} + \bar{p}_b^{c4}|) \qquad (3.7)$$

where $\bar{\boldsymbol{p}}_{\boldsymbol{b}}^{\boldsymbol{C3}}$ and $\bar{\boldsymbol{p}}_{\boldsymbol{b}}^{\boldsymbol{C4}}$ are average power of sub-band b ∈ {δ, θ, α} in channel C3 and C4 respectively. Similarly, calculate relative back hemisphere power RPR(b)bh the difference between O1 and O2 for each sub band in (3.8):

$$\left(|\bar{p}_b^{O1} - \bar{p}_b^{O2}|\right) \, / \, \left(|\bar{p}_b^{O1} + \bar{p}_b^{O2}|\right) \qquad (3.8)$$

where $\bar{\boldsymbol{p}}_{\boldsymbol{b}}^{\boldsymbol{O1}}$ and $\bar{\boldsymbol{p}}_{\boldsymbol{b}}^{\boldsymbol{O2}}$ These values represent the average power of sub-band b (δ, θ, α) in channels O1 and O2.

From the above Eq (3.6, 3.7, 3.8) calculate the numbers of features is 18 attributes. all features can be summarized in Table 3.2 Feature re- arrangement.

To create a model, RFE first eliminates characteristics (features) iteratively before focusing on the ones that remain. To determine which characteristics (and feature combinations) are most important for predicting the target attribute, the model's accuracy is used. The 22 characteristics with the highest rankings (1, true) are the most important to the model's accuracy. All characteristics in the time domain are chosen, along with 16 features in the frequency domain, most of which are associated with the Alpha and Delta bands [22].

Table 3.2 features summarized

Domain type	**Description**	**Feature**	**No. of attribute**
Time domain	**DFA (Detrended Fluctuation Analysis) (4)**	**h(q)c1, h(q)c2, h(q)c3, h(q)c4,**	**4**
Time domain	**Mean value for each ch** **Std value for each ch** **Variance for each ch**	**$\mathbf{Mean_{c1}}$, $\mathbf{std_{c1}}$, $\mathbf{var_{c1}}$** **$\mathbf{Mean_{c2}}$, $\mathbf{std_{c2}}$, $\mathbf{var_{c2}}$** **$\mathbf{Mean_{c3}}$, $\mathbf{std_{c3}}$, $\mathbf{var_{c3}}$** **$\mathbf{Mean_{c4}}$, $\mathbf{std_{c4}}$, $\mathbf{var_{c4}}$**	**12**
Frequency domain	**Relative Power Ratio (Sub band) (12)**	$\bar{P}^{c1}_{\alpha}, \bar{P}^{c1}_{\theta}, \bar{P}^{c1}_{\delta},$ $\bar{P}^{c2}_{\alpha}, \bar{P}^{c2}_{\theta}, \bar{P}^{c2}_{\delta},$ $\bar{P}^{c3}_{\alpha}, \bar{P}^{c3}_{\theta}, \bar{P}^{c3}_{\delta},$ $\bar{P}^{c4}_{\alpha}, \bar{P}^{c4}_{\theta}, \bar{P}^{c4}_{\delta},$	**12**
Frequency domain	**Relative power ratio (hemisphere)**	$\mathrm{RPR}(\delta)_{fh}, \mathrm{RPR}(\delta)_{bh},$ $\mathrm{RPR}(\theta)_{fh}, \mathrm{RPR}(\theta)_{bh},$ $\mathrm{RPR}(\alpha)_{fh}, \mathrm{RPR}(\alpha)_{bh},$	**6**

Here show the pcc.py source python program to plot the welch power density and power band for delta frequency band from 1 to 4 Hz, that listed in Appendix A the output of the program shows in Figure 3.8

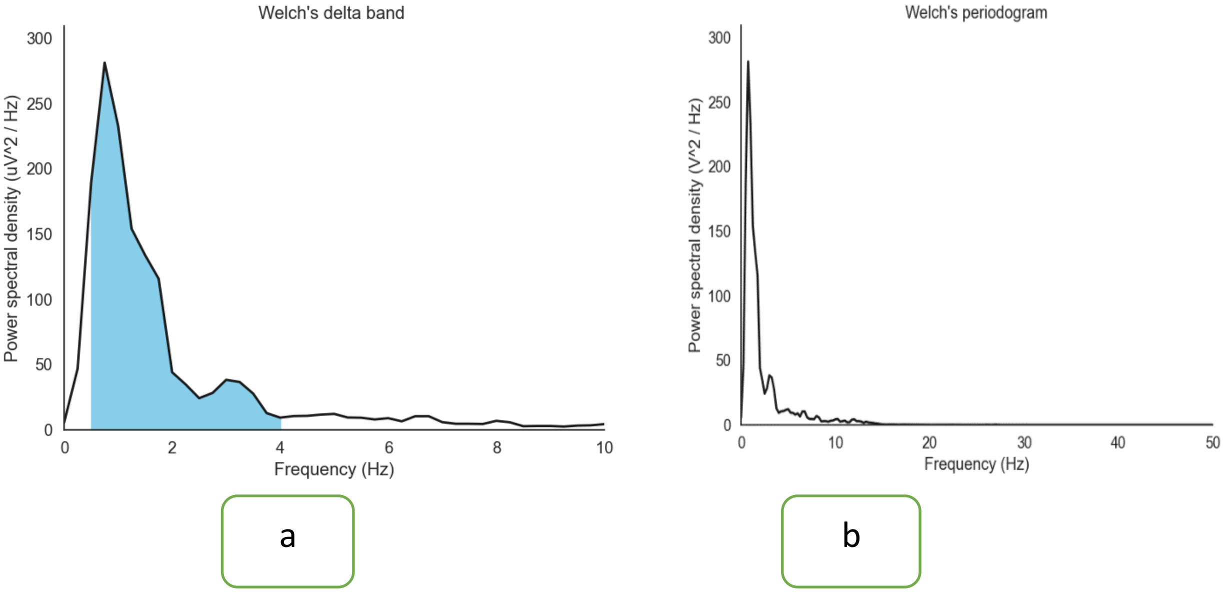


Figure 3.8 display two feature attributes (a) refer to delta band frequency from 1-4Hz (b)Refer to welch spectrum diagram power density

3.9 Classifiers Algorithm Types

There are four distinguishing factors. We use the feature set from Table 3.2 to train a Multi-Layer Perceptron (MLP), a Random Forest, and two bootstrapping models: a Decision Tree and an Extra Tree.

3.9.1 A Decision Tree

Employs a tree-like model of actions and their potential implications, such as chance event outcomes, resource costs, and utility, to help users make better decisions. This is one representation for an algorithm consisting entirely of if/then statements. Though originally developed for use in decision analysis, decision trees have now found widespread use in both operations research and machine learning.

Decision trees (DT) use data attributes to develop basic decision rules to anticipate a target variable's value. Each internal node represents a "test" on an attribute (e.g., whether a coin flip comes up heads or tails), each branch represents the test result, and each leaf node represents a class label. Paths from root to leaf reflect categorization criteria.

A decision tree and impact diagram are used to determine the anticipated values (or utility) of competing options in decision analysis [22].

3.9.2 Extra-Tree Classifier

In heavily randomized trees, split calculations are even more unexpected. Like random forests, a random subset of candidate features is utilized, but instead of finding the most discriminative thresholds, each candidate feature's threshold is picked at random, and the best threshold is used as the splitting criteria. Extra-Tree, like Random Forests, uses several decision trees to predict. Few differences exist between additional and random forests.

Extra-Tree have value, especially when computational cost is a concern. Specifically, when building models that have substantial feature engineering/feature selection pre-modelling steps, and computational cost is an issue Extra-Tree would be a good choice over other ensemble tree-based models.

- Compared to Random Forest, Extra-Tree Classifier uses randomization to minimize variance and computational cost in ensemble tree-based machine learning.

- Extra-Tree Classifier can be used for classification or regression, in scenarios where computational cost is a concern and features have been carefully selected and analyzed.

- Extra-Tree can be implemented from Scikit-learn. The three hyperparameters important for tuning are max_feature, min_samples_leaf, and n_estimators. [81].

3.9.3 MLP is a Multi-Layer Perceptron.

Multilayer perceptron’s (MLPs) are fully linked feedforward artificial neural networks. MLP may refer to any feedforward ANN or to networks with many layers of perceptions (with threshold activation). A multi-layer perceptron with one hidden layer is called a "vanilla" neural network.

An MLP has three node layers: input, hidden, and output. Each node is a nonlinearly activated neuron except the input nodes. MLP trains using backpropagation. Multiple layers and non-linear activation separate MLP from linear perceptron’s. It can identify non-linearly separable data [86]

MLPs are feedforward artificial neural networks that output from inputs. An MLP has a directed graph with input and output layers and many input node layers. To train networks, MLP backpropagates. Deep learning MLP.

A directed graph with several layers is a multilayer perceptron with one-way node signals. All nodes except input have nonlinear activation functions. The MLP uses backpropagation for supervised learning. MLP is deep learning because to its many neurone layers.

Supervised learning problems are often solved with MLP in computational neuroscience and parallel distributed computing. Machine translation, speech, and image recognition are apps. [85]

3.9.4 Random Forest (RF)

Random forests—also termed random choice forests—are ensemble learning approaches for classification, regression, and other applications. This method works. Because training creates several decision trees. A random forest outputs the category most trees chose in classification problems. Regression problems yield each tree's "mean" or "average" forecast. Random decision forests handle training set overfitting. Random forests are more accurate than gradient-boosted trees but less accurate than choice trees. However, data feature performance may suffer [22].

Random forest is a popular data science algorithm. Random Forest-supervised machine learning techniques are used in classification and regression. It creates decision trees from several data, categorizes by majority vote, and regresses by average [87]

Leo Breiman and Adele Cutler created random forest, a versatile machine learning technique. It uses several decision trees to predict or classify. Integrating several tree outputs yields a more accurate and consolidated result in the random forest method. [87]

The software solves classification and regression issues simply and adaptably, making it popular. The algorithm's ability to handle complicated datasets and prevent overfitting makes it ideal for many machine learning prediction challenges. [86]

3.10 Feature Extraction Process Flow Diagram

Show the flow operation block diagram with the original Python software using PYTHON-MNE types. It is unique to Python since it handles electroencephalograms. Block diagram shows feature extraction processes. You may also input machine learning from here. The true Python script used in this study is attached. Python 3 and PyCharm 3 can execute 99.py [88]. Python code (feature 99.py) can retrieve all values and depicts the genuine procedure.

All characteristics are source code, and Appendix A has the list CODE (feature 99.py) and Figure 3.9 feature extraction block diagram and Figure 3.10(a, b) flowchart of feature extraction programmed operation.

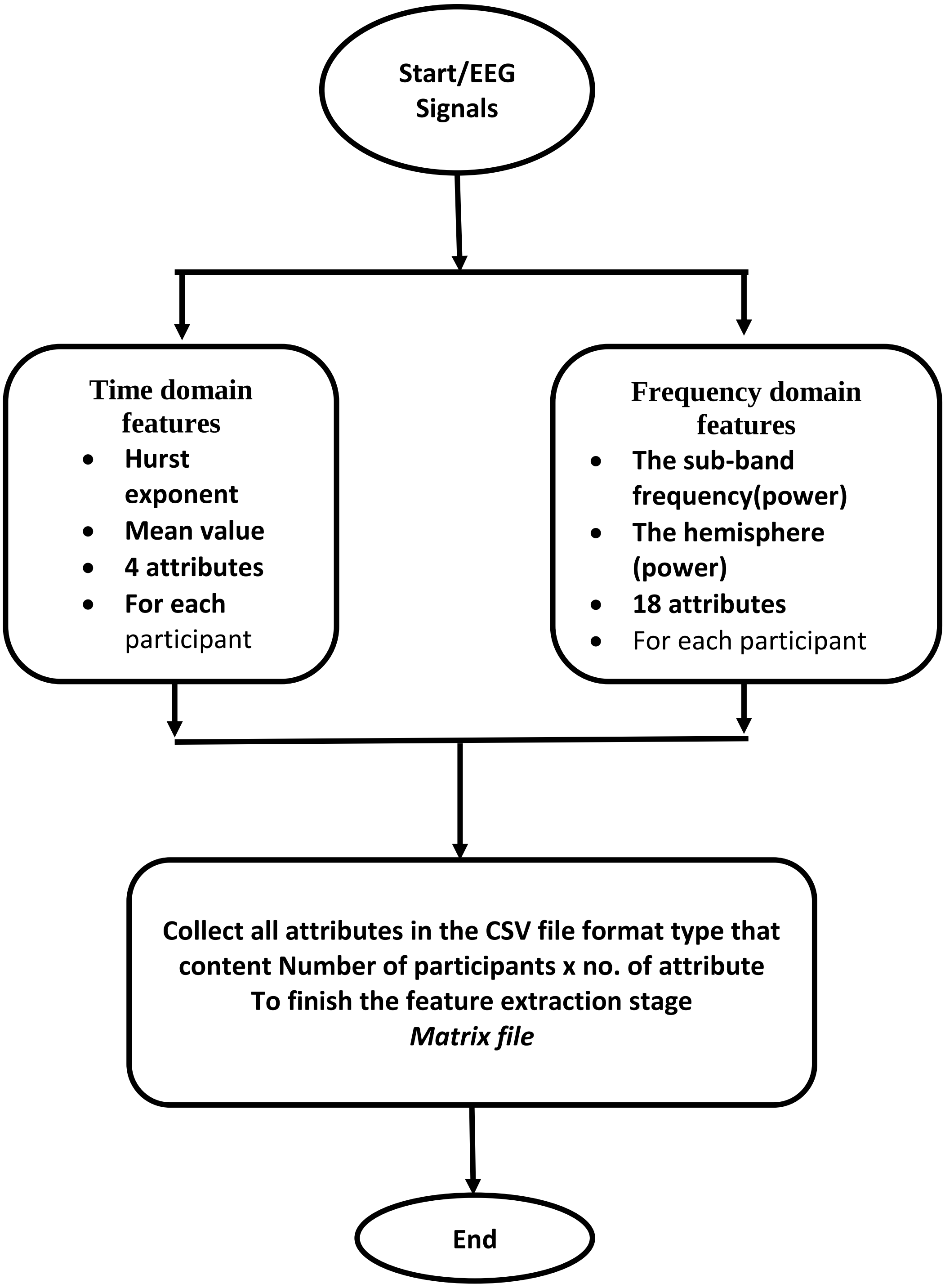

Start/EEG Signals
Time domain features
Hurst exponent
Mean value
4 attributes
For each participant
Frequency domain features
The sub-band frequency(power)
The hemisphere (power)
18 attributes
For each participant
Collect all attributes in the CSV file format type that content Number of participants x no. of attribute
To finish the feature extraction stage
Matrix file
End

Figure 3.9 The Feature Extraction Block Diagram

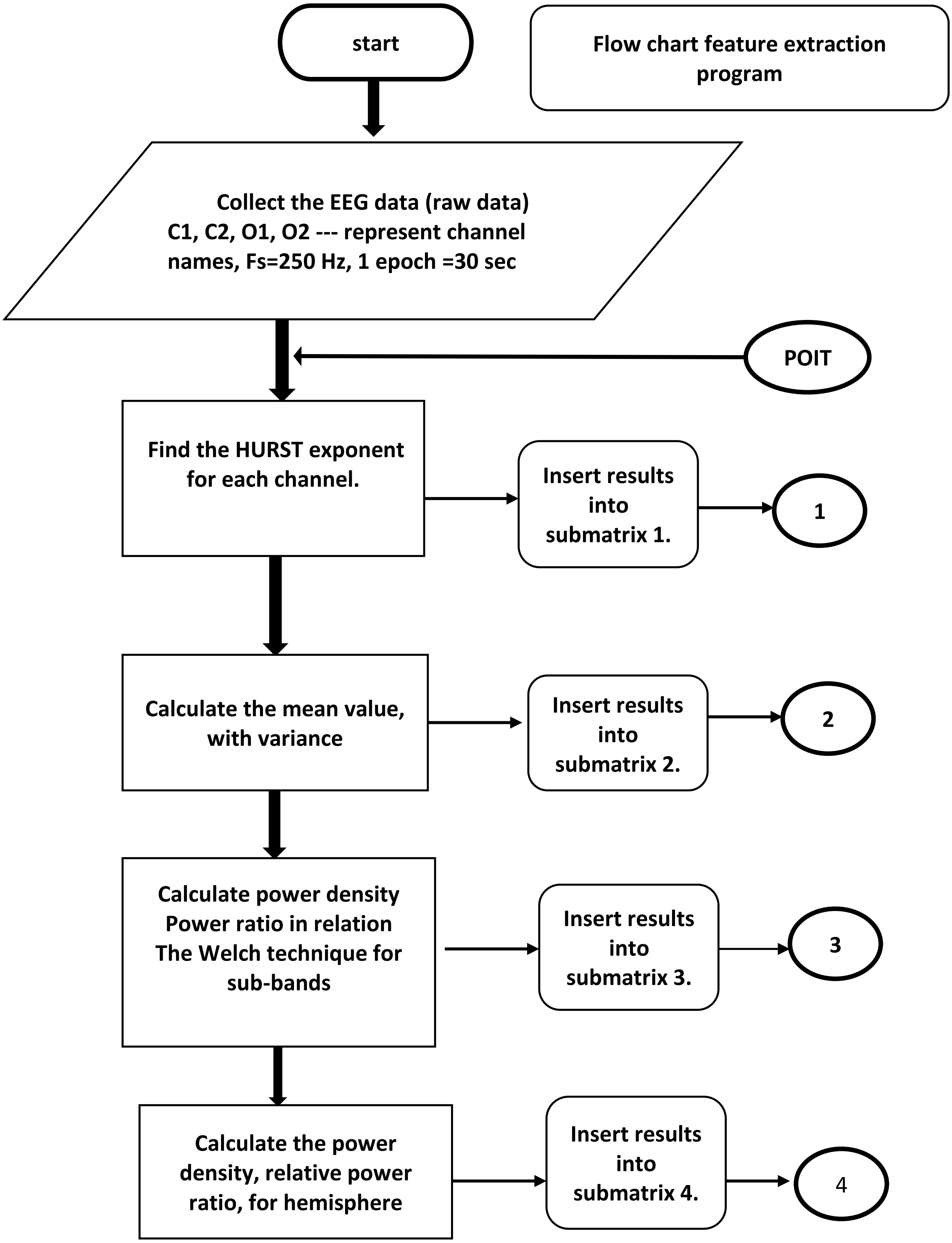

start
Flow chart feature extraction program
Collect the EEG data (raw data)
C1, C2, O1, O2 --- represent channel names, Fs=250 Hz, 1 epoch =30 sec
POIT
Find the HURST exponent for each channel.
Insert results into submatrix 1.
1
Calculate the mean value, with variance
Insert results into submatrix 2.
2
Calculate power density Power ratio in relation The Welch technique for sub-bands
Insert results into submatrix 3.
3
Calculate the power density, relative power ratio, for hemisphere
Insert results into submatrix 4.
4

Figure 3.10 (a) flowchart operation of the EEG feature extraction program

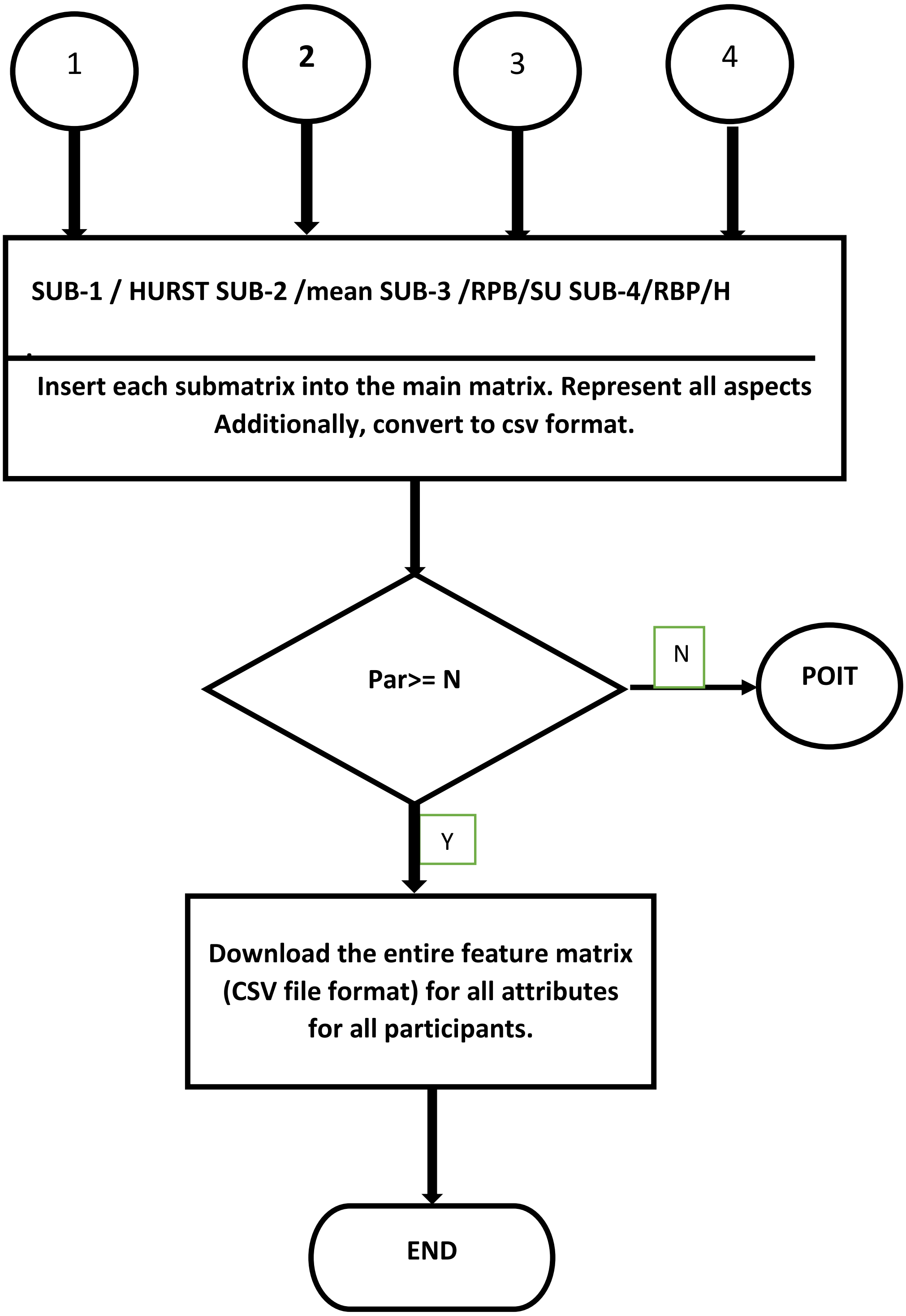

1
2
3
4
SUB-1 / HURST SUB-2 /mean SUB-3 /RPB/SU SUB-4/RBP/H
Insert each submatrix into the main matrix. Represent all aspects Additionally, convert to csv format.
Par>= N
N
POIT
Y
Download the entire feature matrix (CSV file format) for all attributes for all participants.
END

Figure 3.10 (b) flowchart operation of the EEG feature extraction program

3.11 Second technique for image processing

The above route detailed the time series path approach to extract EEG signal features. The date was obtained from al-Hilla Teaching Hospital[89]and al Imam al Sadiq Teaching Hospital [90] in that bath. MRI is utilised to identify early stroke warnings in the future. Data was gathered from the gadget as follows:

1) CT scans are generally utilised in medical procedures to see inside body components. Computer-controlled X-ray detectors and sources provide a tomography-type picture. It processes data and creates a CT scan for cross-sectional imaging of ischemic and hemorrhagic strokes.
2) *Magnetic resonance imaging (MRI)* is a type of scan that uses strong magnetic fields and radio waves to produce detailed images of the inside of the body. An MRI scanner is a large tube that contains powerful magnets. In that path the image processing is a power full method that used to achieve the goal, the above device can be used to collect data as an image input to the proposed system that suggest hear.

The proposed algorithm uses Grayscale Image conversion, Normalization, Image filtering, segmentation, Morphological Operation, and calculation based on the connected component to detect the brain strokes. As shows in the following block diagram Figure 3.11 represent the procedure steps in the IP path.

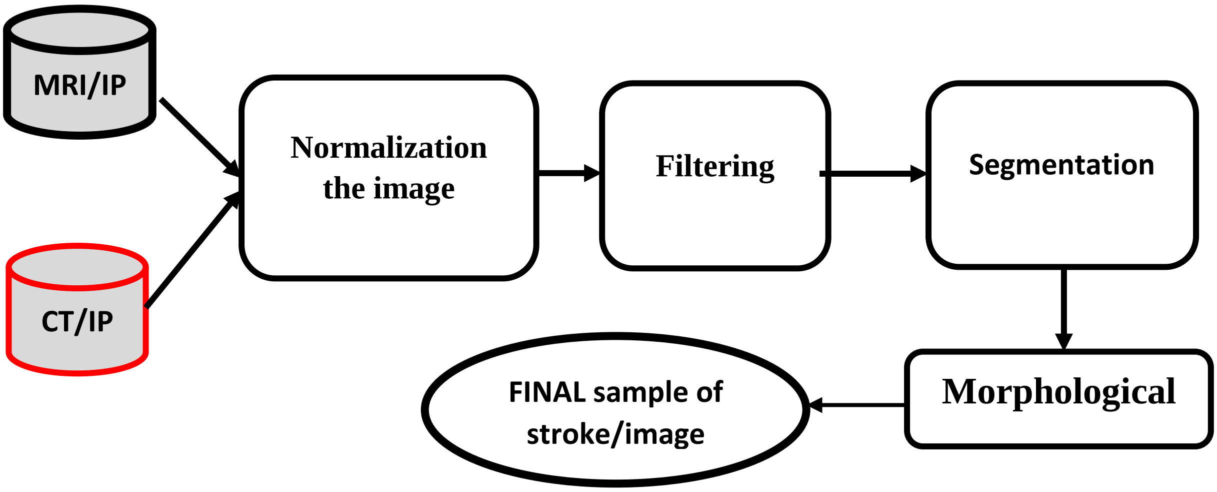


Figure 3.11 depicts the phases of image processing in that work.

Input to the Algorithm: A computerized tomography (CT) Scan image/ Magnetic Resonance Imaging (MRI.)

Output: Stroke portion detection on the provided image.

1) **In the first step**, a gray scale or a colored image is provided to the system as an input. In the next step, a colored image is converted into the gray scale image. Weighted sum of Read, Breen, Blue, RGB component is used to carry to this conversion. Gray image conversion is carried out using equation 3.9:

$$\mathbf{Y = 0.2126R + 0.7152G + 0.0722B} \qquad \mathbf{(3.9)}$$

An output of this step is gray scale image with eliminated hue and saturation, while luminance is retained to get expected output. Figure 3.12 shows the brain stroke in the gray scale image [91].

2) *After generation* of gray scale image to maintain the uniformity in sizing, normalization is performed. In this step, images are scaled into 300 x 280 using normalization process. Depending on the current size, normalization either reduce, increase, or retain the image size of the output image. Equation 3.10 refer to normalization step.

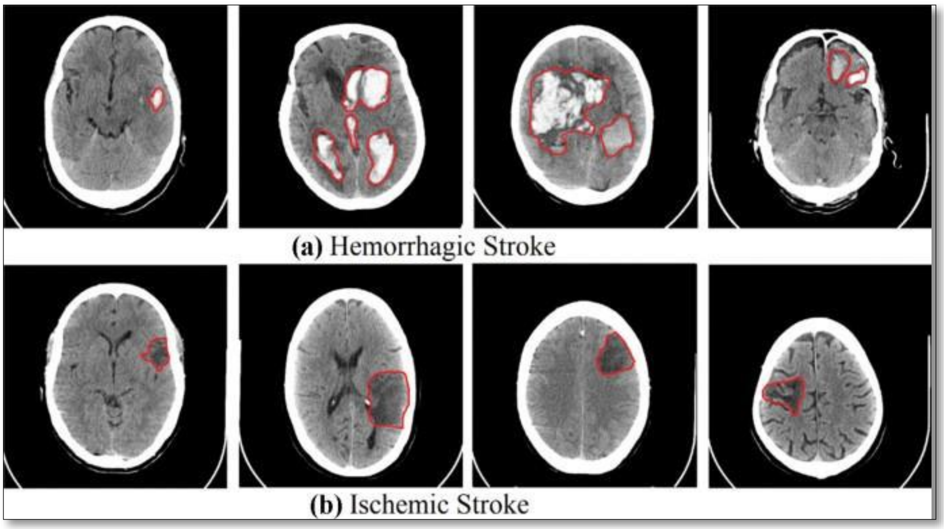


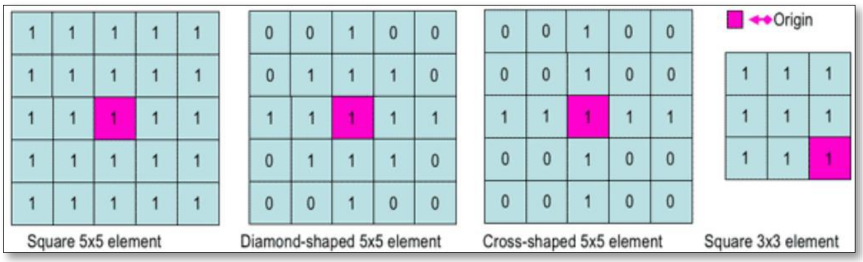
Figure 3.12 Brain stroke in the gray scale image

$$\mathbf{Normlized(I)} = (\boldsymbol{I} - \boldsymbol{Min})\left(\frac{\mathbf{300-280}}{\boldsymbol{Max-Min}}\right) + \mathbf{280} \qquad \textbf{(3.10)}$$

3) Filtering is applied on the normalized image for feature enhancement. Filtering is used to emphasize or remove certain attributes of an image. This process includes image smoothing, sharpening, noise removal and enhancement of edges. In this research, Low pass filter is used for smoothing an image, while high pass filers are applied for edge detection and sharpening.
4) Segmentation. Many techniques use in this step, thresholding method is used. In which image is converted into the binary image. In this image intensity is transformed into the binary image. In this method iteration is carried out through all the possible values of threshold. Based on this the calculation of spread for the pixel levels each side of the threshold is carried out[91]. Otsu's thresholding Algorithm:
5) Transformations Morphological. Morphological transformations are straightforward image-shape-based operations. Typically, it is performed on binary images. It requires two inputs: the original image and a structuring element or nucleus that determines the operation's nature. Erosion and dilation are fundamental morphological operators. [92]
6) The Otsu method, named after Nobuyuki Otsu, is utilised for autonomous image thresholding. In its most basic form, the algorithm returns a single intensity threshold that divides pixels into foreground and background classifications. This threshold is determined by minimising intra-class intensity variance and maximising inter-class variance, respectively. Otsu's method is a one-dimensional discrete analogue of Fisher's Discriminant Analysis related to Jenkins' method of optimisation and is equivalent to a globally optimal means conducted on the intensity histogram. The original paper described the extension to multi-level thresholding, and computationally efficient implementations have since been proposed. It includes the following details: [93]

1) Histogram and probabilities for each intensity level is computed.
2) Initial µI (0) and ɯI (0) setup, where ɯ is the probabilities of classes separated by a threshold and µ is class mean.
3) Loop through every possible threshold value from 0 to max intensity
   - Update µi and ɯi values
   - Calculate variance σ2(t)
4) Intended threshold corresponding to the maximum σ2(t)

After threshold segmentation using Otsu's thresholding Algorithm, few morphological operations are carried out. An aim of this step is to identify only parts of image which contain stroke signals. This operation takes image and structuring elements as an input. Structuring elements are nothing but set of co-ordination point. In a morphological operation, each image pixel is corresponding to the value of other pixel in its neighborhood. Structuring elements do not have specified shape, it can be of any shape and structuring element is the one which estimates the accurate effects of morphological operation on an image. The pattern of 1’s and 0’s specifies the shape of the structuring element. Figure 3.13 shows the example of structuring elements. [94].refer to MM processing.



Square 5x5 element Diamond-shaped 5x5 element Cross-shaped 5x5 element Square 3x3 element

Figure 3.13 Examples of simple structuring elements

For stroke portion detection, initially a white region from the image needs to be recognized. Spatial structure is used to recognize the object. Using connectivity, neighborhood and region boundaries connected components are discovered. Black pixels or 0’s is used to generate the region boundaries. Using these structuring elements white region is analyzed and stroke portion is detected [95]

After Normalization, Median filtering, segmentation and removing small blobs, stroke portion can be detected as illustrated in Figure 3.14 below.

The morphology method proved effective in this study.

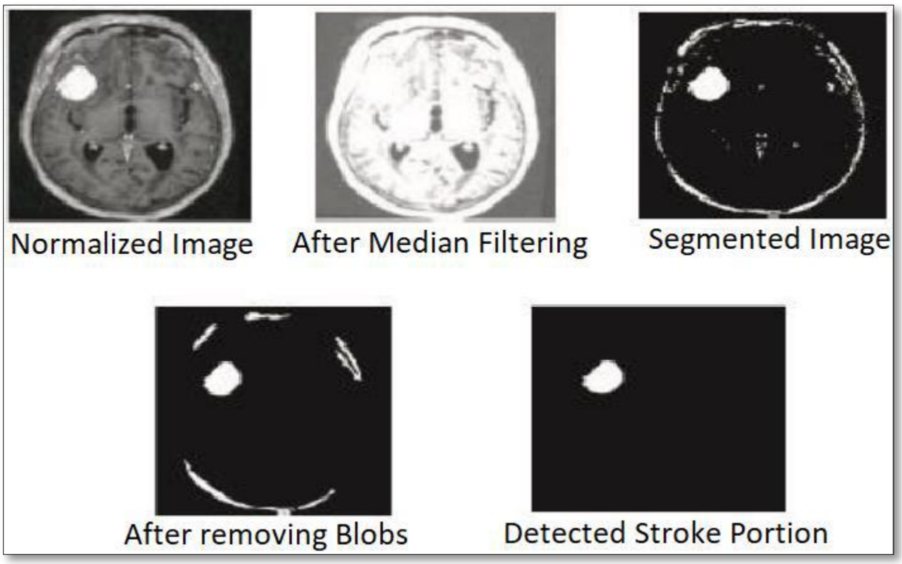


Figure 3.14 transition of CT scan image to Detect Stroke

Use set theory, lattice theory, topology, and random functions to study and work with topological shapes. This is what mathematical morphology (MM) is all about. Most of the time, MM is used on digital pictures, but it can also be used on graphs, surface meshes, solids, and a lot of other types of spatial structures [96].

in that part use four algorithm to test the proposed method show in above steps, random forest, K-nearest neighbor, ANN, Proposed Algorithm, the proposed algorithm get good result[92],the results show in chapter four

Steps of the Proposed Method

Input: A computerized tomography (CT) scan image

Output: Image with detected brain strokes

Step 1: Scan and process the grayscale or color image.

Step 2: RED, GREEN, and BLUE RGB component of an image is computed.

The weighted average of RGB is found using the method in Eq. 3.9 so that it can be turned into a grayscale picture. Change the RGB picture component that is currently there to an estimated component [95].

Step 3: Normalization of an image is performed using Eq (3.10)

Step 4: In this research, Low pass filter is used for smoothing an image, while high pass filers are applied for edge detection and sharpening. Store the filtering data into Array. Output of this step is consequential enhanced image [95].

Step 5, the thresholding technique is employed for feature detection. the picture goes through the binary image conversion process. Otsu's thresholding approach is implemented for this function. As was said before, this algorithm is put to use when setting thresholds. The intensity picture acquired in the preceding step may be transformed into a binary image by setting the threshold to a value between 0 and 1. As a result, there is less variation within the white and black pixel classes.

Step 6: After threshold using Otsu's thresholding Algorithm, few morphological operations are carried out. An aim of this step is to identify only parts of image which contain stroke signals. In a morphological operation, each image pixel is corresponding to the value of other pixel in its neighborhood[97].

Step 7: This is where you figure out the linked components. An important job for finding strokes quickly is finding white areas. A study of the regional organization is done using white area identification. To find related components, you need to know about region boundaries and connection.

You can split a binary picture into black and white areas, and you can find blobs in white areas.

Step 8: Finally, the size and boundaries of each blob are found. Blobs with an area of less than 100 are taken away. The center and width are then saved in the collection. A cutoff number tells the machine whether the blob is a real stroke, or a fake posit.

Chapter Four

Results and Discussion

Chapter four

Results and Discussion

4.1 Introduction

The outcome of this study may be divided into two primary components, depending on the Bese of the analysis of signals; in that work, the inquiry into signals and the analysis of signals are both included.

- Time series analysis to the EEG
- The image processing (IP) analysis with more technique, using MRI path.

4.2 The Electroencephalography (EEG) Based results.

In this specific area of work, EEG readings were needed to do all the important research and processing, as well as to acquire all of the characteristics required to complete all of the data collection. After that, starting to analyze the results using many different methods, and then evaluate and contrast the various ways. This idea is shown by the functional graphic Figure 4.1 that comes up next. The general block diagram represents the flow of operation of processing with 80 participants (par)and 26 attributes.

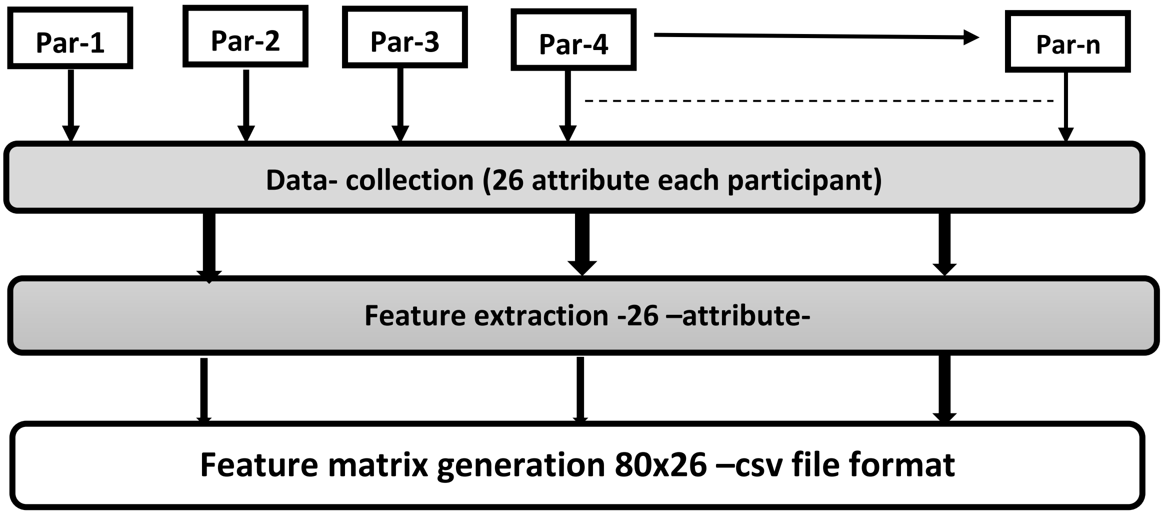

Par-1
Par-2
Par-3
Par-4
Par-n
Data- collection (26 attribute each participant)
Feature extraction -26 –attribute-
Feature matrix generation 80x26 –csv file format

Figure 4.1 Block diagram of the operation of processing with EEG

4.2.1 Data Description with Attributes

At first limit the number of the participants 80 persons only, using an electroencephalography (EEG) path, with different numbers of signals that depended on the recorder system used with each case.

- Fix sampling frequency to all participants at 250Hz
- Fix the number of channels for each participant to 6 channels only.
- Selecting the location of channels in the front and behind the head, depends on the stroke detection.
- Limit the time to 60 sec, this is to conserve memory size.

These are the headline that must be used to collect the data, all the above data get from the location [69] [70]

- For each participant select 6 channels and get 26 attributes
- These attributes distributed into two domains.
    - ***For time domain***:
- 8 attributes located in the time domain (4, Hurst-exponent,4 mean value) each channel represented by 4 attributes (2, for Hurst, 2 for mean.)
    - ***For frequency domain***:
- 18 Attributes located in the frequency domain and distribution as follows.
- 12- attribute represent *Relative Power Ratio (RPR)* band power for 3 bands (alpha, theta, delta) each band get 3 attributes for each channel, gets 12 attributes for each participant, this result from Eq 3.6 in Chapter 3 with the python code,
- 6-Attribute represents Relative Power Ratio (RPR_{FH}) for (hemisphere) 3 band (Alpha, Theta, Delta) it represents the low frequency bands,2 attribute for each band, it contains two group front and back of the head. It can be gets from Eq. 3.7 in chapter 3,

Similarly, calculate the relative *back hemisphere* power (RPR_{BH}) the difference between O1 and O2 for each sub, Eq 3.7 in character 3.

there are 18 features for each epoch in the frequency domain.[98], After the above steps of operation, produced a matrix 80x26 that represents the feature matrix of all data . Now at this point get right into the *machine learning* section, by collecting the data in matrix format and transfer into CSV file format with the help of a Python language program and applying multi algorithms focusing on classification and prediction into the main goal (stroke detection). then illustrate the data file with some plot files to Group of participant's *Table 4.1(data table),* the table include four participants only and arranged other participants in Appendix BN, data Appendix.

Table 4.1 (A, B) Contains 26 Attributes for Four Participants

	Col-1	Col-2	Col-3	Col-4	Col-5
A	**Attributes**	**PAR-1**	**PAR-2**	**PAR-3**	**PAR-4**
1	**Hurst-1**	**0.202647**	**0.202647**	**0.202647**	**0.202647**
2	**Hurst-2**	**0.202647**	**0.202647**	**0.202647**	**0.202647**
3	**Hurst-3**	**0.202647**	**0.202647**	**0.202647**	**0.202647**
4	**Hurst-4**	**0.202647**	**0.202647**	**0.202647**	**0.202647**
5	**Mean -1**	**1.21204**	**7.440866**	**1.18913**	**1.18913**
6	**Mean -2**	**2.440866**	**2.440866**	**2.440866**	**2.440866**
7	**Mean -3**	**1.18913**	**1.18913**	**1.18913**	**1.18913**
8	**Mean -4**	**-1.760699**	**-1.560699**	**-1.560699**	**-1.560699**
9	**RPR-1**	**0.54877**	**0.64877**	**0.64877**	**0.64877**
10	**RPR-2**	**0.39019**	**0.35019**	**0.35019**	**0.35019**
11	**RPR-3**	**0.54877**	**0.58877**	**0.58877**	**0.58877**
12	**RPR-4**	**0.39019**	**0.39019**	**0.39019**	**0.39019**
13	**RPR-5**	**0.54877**	**0.44877**	**0.44877**	**0.44877**

Table 4.1.B

	Col-1	Col-2	Col-3	Col-4	Col-5
B	Attributes	PAR-1	PAR-2	PAR-3	PAR-4
14	RPR-6	0.10915	0.12929	0.13556	0.13453
15	RPR-7	0.18436	0.10137	0.17593	0.15539
16	RPR-8	0.05244	0.02767	0.03096	0.04148
17	RPR-9	0.10477	0.02895	0.05758	0.05259
18	RPR-10	0.05244	0.02767	0.03096	0.04148
19	RPR-11	0.20477	0.2895	0.15758	0.35259
20	RPR-12	0.18436	0.10137	0.17593	0.15539
21	RPRfh1	0.074	0.032	0.057	0.053
22	RPRfh2	0.206	0.444	0.088	0.142
23	RPRfh3	0.277	0.315	0.131	0.073
24	RPRbh1	0.011	0.017	0.04	0.023
25	RPRbh2	0.277	0.315	0.131	0.073
26	RPRbh3	0.1211	0.018	0.045	0.032

Note that:

hurst --- refers to hurst exponent attribute in the time domain,

Mean – refers to the mean value of each channel with the participant.

RPR --- refer to relative power ratio (band of frequency),

RPR$_{fh}$, ***RPR***$_{bh}$------ refer to relative power of hemisphere.

There are 26 attributes for each participant –using 4 channels only.

All other DATA are listed with Appendix -B (data values)

- **Attribute description**

From table 4.1(A, B)

NOTE that, per --- refer to the participant the table represent a matrix the row refer to --- attributes.

The A, B, col 1, refer to attributes number with description.

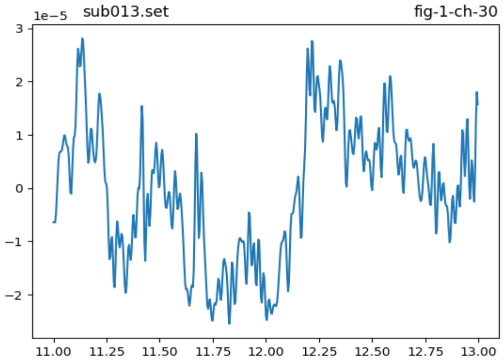
col-1, col-2, col-3, col-4, col-5----- represent the participant number, each column contains 26 attributes (values)

The **Appendix B (DATA APPENDIX -B)** includes forty participants represent the attribute values.

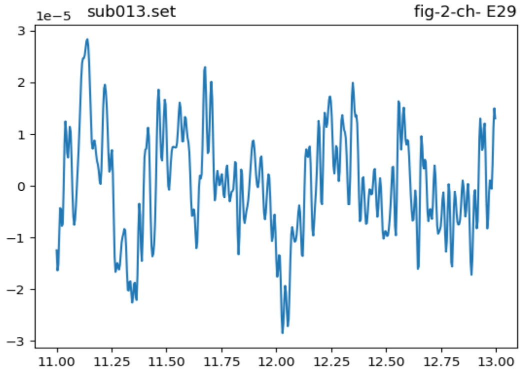
Figure 4.2 shows the plots for *one participant* that includes 26 attributes, from dataset 2691 refer to table 3.1 in chapter three.

Participant no. code. sub0013, fs =250 HZ, into 4 channels. the other plot **(select 4 participants only)** of the participant are collected in Appendix-C. **(Plot appendix).**

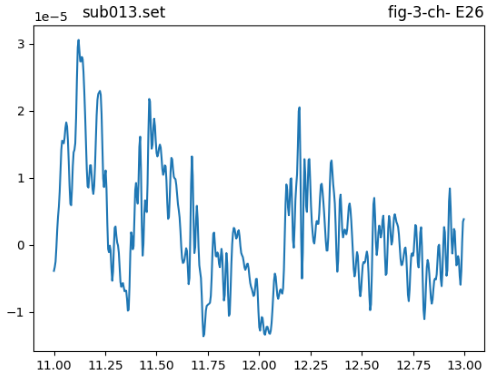
One participant with 26 plots represent all



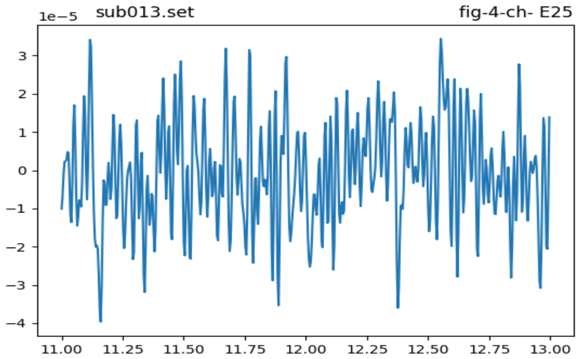
(a



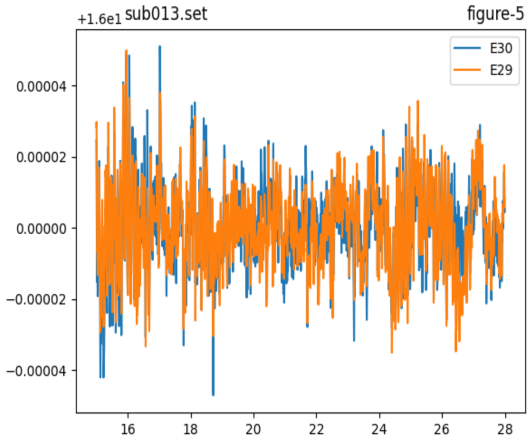
(b)



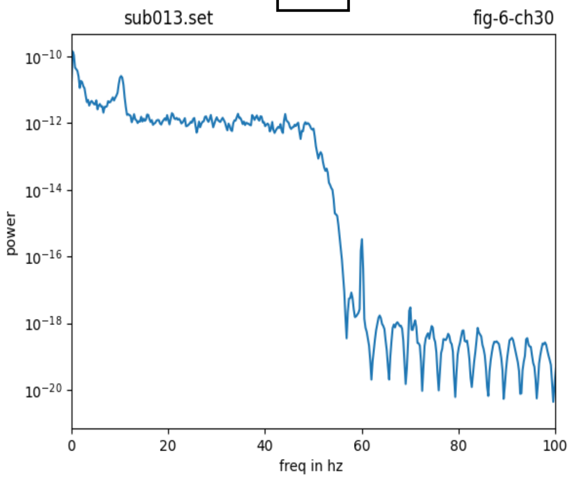
(c)



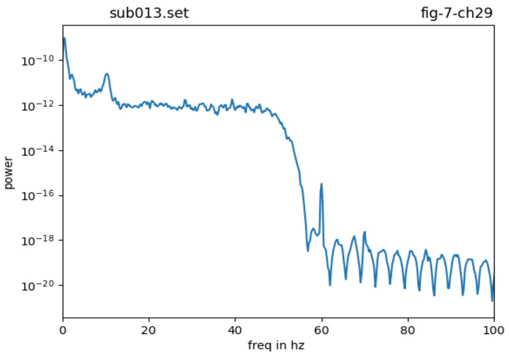
(d



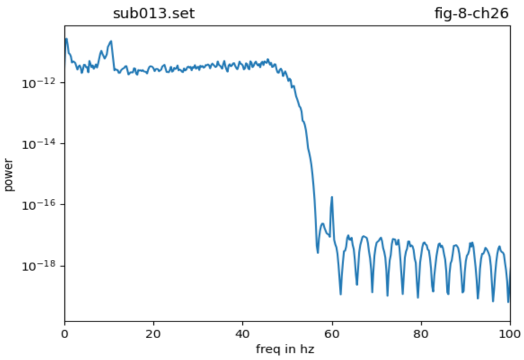
(e)



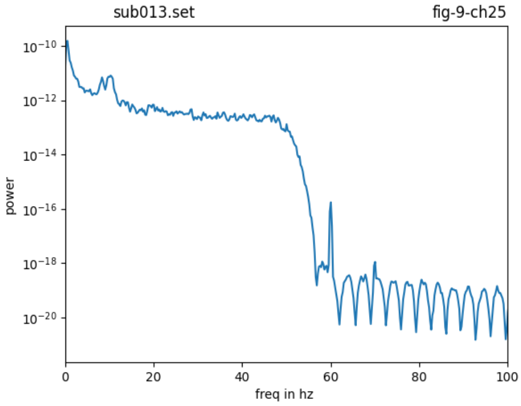
(f)


sub013.set
fig-7-ch29
power
10^{-10}
10^{-12}
10^{-14}
10^{-16}
10^{-18}
10^{-20}
0
20
40
60
80
100
freq in hz

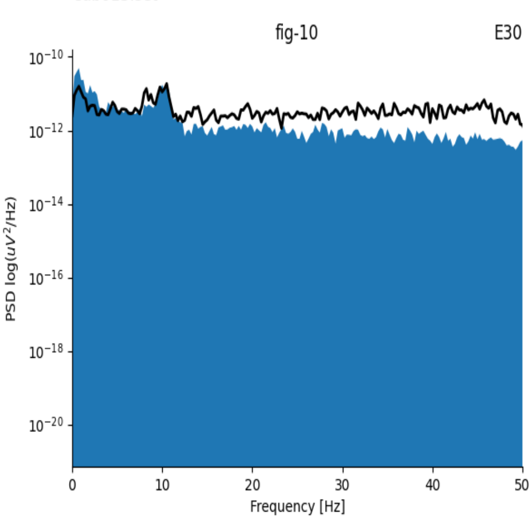
(g)


sub013.set
fig-8-ch26
power
10^{-12}
10^{-14}
10^{-16}
10^{-18}
0
20
40
60
80
100
freq in hz

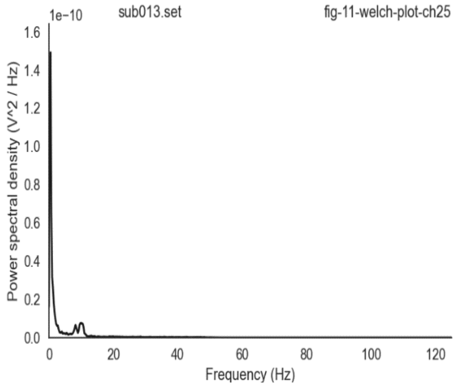
(h)


sub013.set
fig-9-ch25
power
10^{-10}
10^{-12}
10^{-14}
10^{-16}
10^{-18}
10^{-20}
0
20
40
60
80
100
freq in hz

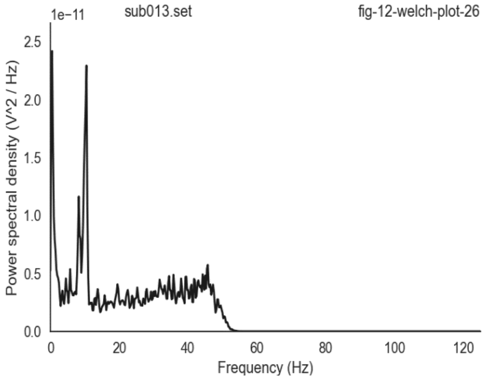
(i)


sub013.set
fig-10
E30
PSD log(uV^2/Hz)
10^{-10}
10^{-12}
10^{-14}
10^{-16}
10^{-18}
10^{-20}
0
10
20
30
40
50
Frequency [Hz]

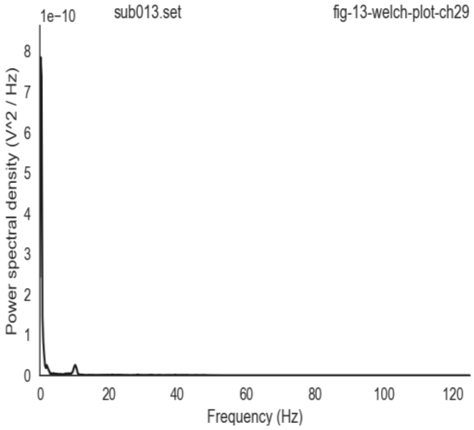
(j)


1e−10
sub013.set
fig-11-welch-plot-ch25
Power spectral density (V^2 / Hz)
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80
100
120
Frequency (Hz)

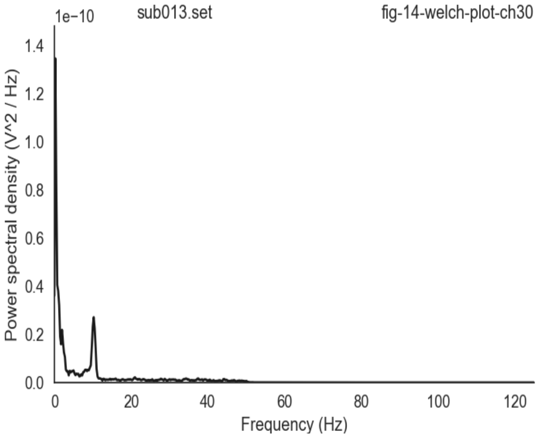
(k)


1e−11
sub013.set
fig-12-welch-plot-26
Power spectral density (V^2 / Hz)
2.5
2.0
1.5
1.0
0.5
0.0
0
20
40
60
80
100
120
Frequency (Hz)

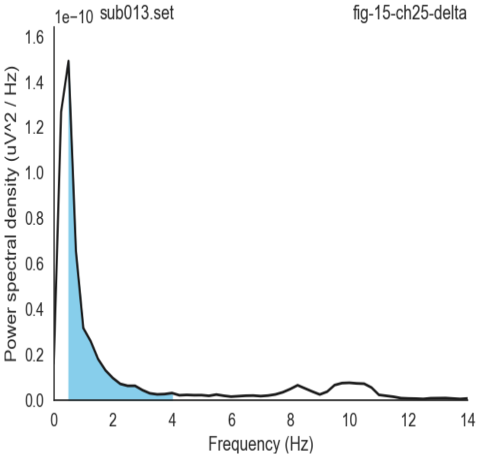
(l)


1e−10
sub013.set
fig-13-welch-plot-ch29
Power spectral density (V^2 / Hz)
0
1
2
3
4
5
6
7
8
0
20
40
60
80
100
120
Frequency (Hz)

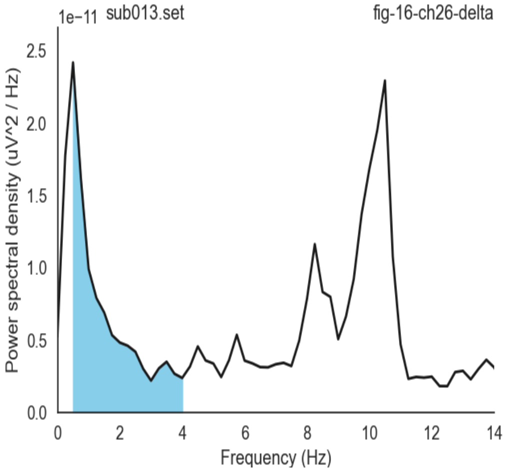
(m)


1e−10
sub013.set
fig-14-welch-plot-ch30
Power spectral density (V^2 / Hz)
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
0
20
40
60
80
100
120
Frequency (Hz)

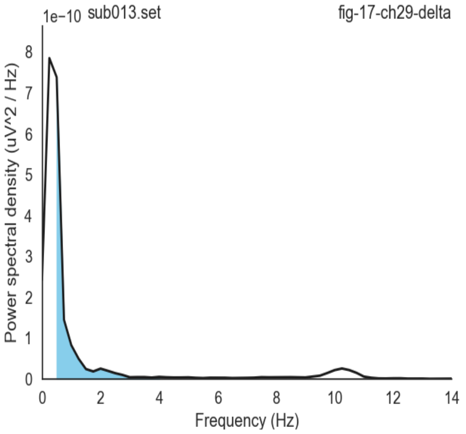
(n)


1e−10
sub013.set
fig-15-ch25-delta
Power spectral density (uV^2 / Hz)
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
0
2
4
6
8
10
12
14
Frequency (Hz)

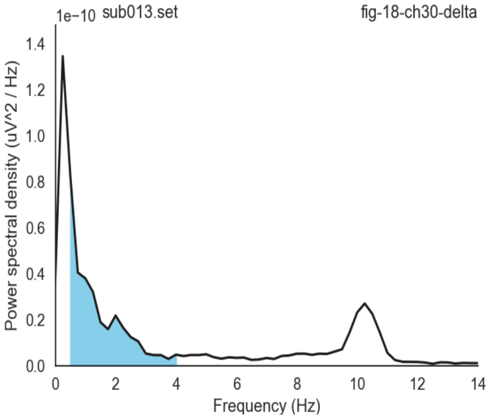
(o)


1e−11
sub013.set
fig-16-ch26-delta
Power spectral density (uV^2 / Hz)
0.0
0.5
1.0
1.5
2.0
2.5
0
2
4
6
8
10
12
14
Frequency (Hz)

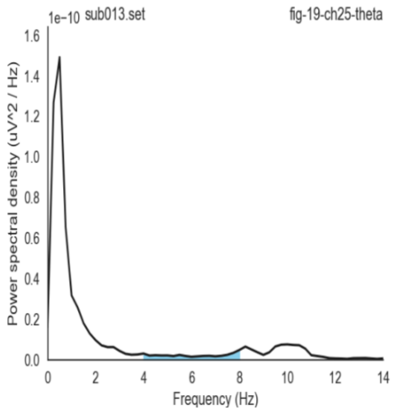
(p)


1e−10
sub013.set
fig-17-ch29-delta
Power spectral density (uV^2 / Hz)
0
1
2
3
4
5
6
7
8
0
2
4
6
8
10
12
14
Frequency (Hz)

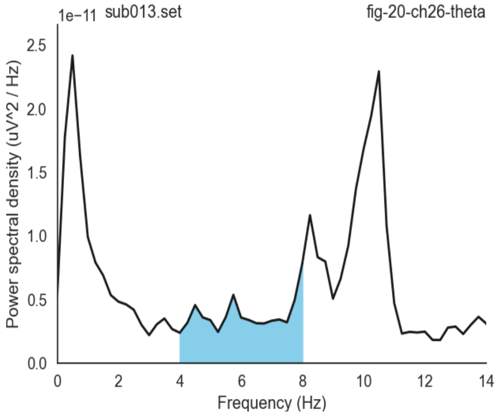
(q)


1e−10
sub013.set
fig-18-ch30-delta
Power spectral density (uV^2 / Hz)
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
0
2
4
6
8
10
12
14
Frequency (Hz)

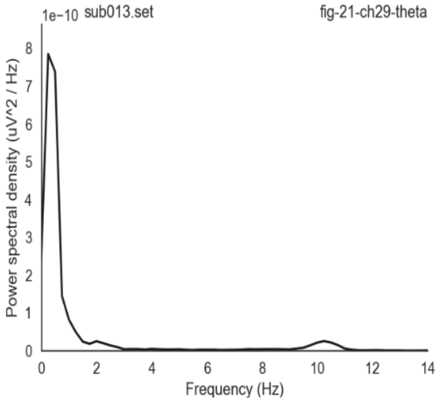
(r)


1e−10
sub013.set
fig-19-ch25-theta
Power spectral density (uV^2 / Hz)
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
0
2
4
6
8
10
12
14
Frequency (Hz)

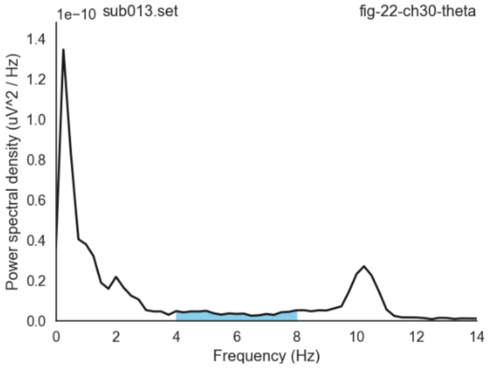
(s)


1e−11
sub013.set
fig-20-ch26-theta
Power spectral density (uV^2 / Hz)
0.0
0.5
1.0
1.5
2.0
2.5
0
2
4
6
8
10
12
14
Frequency (Hz)

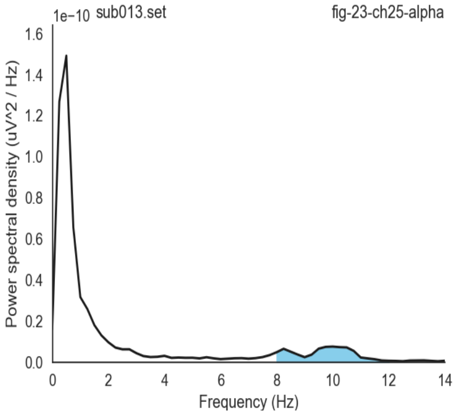
(t)


1e−10
sub013.set
fig-21-ch29-theta
Power spectral density (uV^2 / Hz)
0
1
2
3
4
5
6
7
8
0
2
4
6
8
10
12
14
Frequency (Hz)

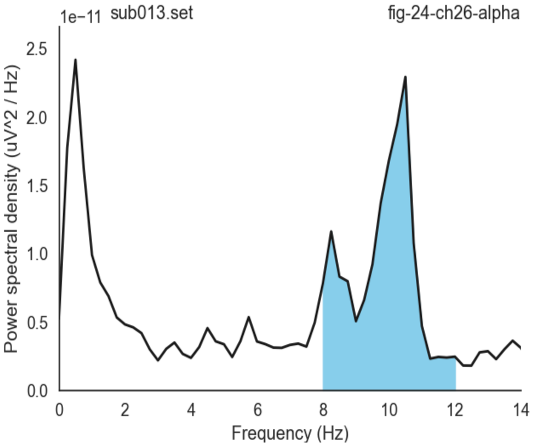
(u)


1e−10
sub013.set
fig-22-ch30-theta
Power spectral density (uV^2 / Hz)
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
0
2
4
6
8
10
12
14
Frequency (Hz)

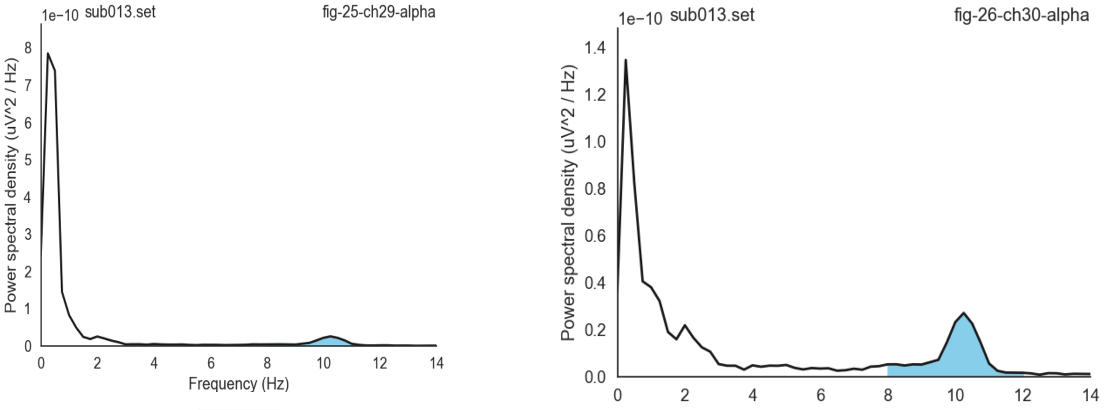
(v)


1e−10
sub013.set
fig-23-ch25-alpha
Power spectral density (uV^2 / Hz)
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
0
2
4
6
8
10
12
14
Frequency (Hz)

(w)


1e−11
sub013.set
fig-24-ch26-alpha
Power spectral density (uV^2 / Hz)
0.0
0.5
1.0
1.5
2.0
2.5
0
2
4
6
8
10
12
14
Frequency (Hz)

(x)



(y)

(Z)

Figure 4.2 plot all attribute for one participant.

Note that: the number of channels 4, with 2 reference channels

- The *plots*. a, b, c, d, e ---- represent the time series of each channels E30-CH30, E29—CH-29, E26-CH26, E25—CH25, to calculate hurst exponent and the mean value of each, *plots* e, f, g, h --- represent power spectrum plot, using Welch method, for long time, *plots* i, j, k, l, m , represent the power spectrum plot, using welch, for fixed window 4 sec, plots o, r, s, t….y, z represent the band-power delta, theta, alpha, and hemisphere relative power ratio from the Figure 4.1 (block diagram), and refer to Figure 3.5 in chapter three when gets feature 80x26, from this point now change the direction of the work to machine learning part.

4.3 Algorithms That Used in this work

- **Multi-layered perception (MLP):**Multi-layer perceptron. It has three levels: an input layer, one or more hidden layers, and an output layer as the last. Each layer save the output layer has a bias neuron and is totally connected to its predecessor. This model is sensitive to hyper-parameters, hence the grid function was utilized to find optimum values for the bootstrap models in this research.[22]

- **Decision Tree (DT):** The objective of Decision Trees (DT) is to construct a model that is capable of predicting the value of a target variable via the discovery of simple decision rules that are inferred from the characteristics of the data. [22].
- **Extra-Tree (ET):** The Extra-Tree (ET) approach for calculating splits in highly randomized trees take randomization to the next level. As in random forests. This approach uses a random subset of candidate features. Instead of finding the thresholds with the greatest discrimination, thresholds are generated at random for each candidate feature and used as the splitting criteria. [22]
- **Random forest (RF):** Leo Breiman is credited with the invention of Random Forests. He was motivated to do so by the previous work done by Amit and Geman. Random Forests are an outgrowth of Breiman's bagging notion, although this is not evident from the explanation, and they were created as a rival to boosting, even though this is not clear from the description [99].

Random Forests may be used for either a categorical response variable, which is referred to as "classification" in, or a continuous response, which is referred to as "regression". Both of these applications are discussed. In a similar manner, the predictor variables may either be categorical or continuous in nature [99].

- **Random Forests are appealing from a computational perspective for the following reasons:**
  - they naturally handle both regression and (multiclass) classification.
  - they are relatively quick to train and to predict.
  - they are dependent on only one or two tuning parameters.
  - they have an estimate of generalization error that is built in.
  - they can be used directly for high-dimensional problems.
  - they can be easily implemented in parallel.

Random Forests are intriguing from a statistical standpoint because to the extra capabilities that they provide, such as:

- measures of variable significance.
- differential class weighting.
- missing value imputation.
- and visualization.

4.4 some important concepts

The criteria that were utilized to get the result were employed before discussing and displaying the results and comparing them to one another.

a) the receiver operation curve (ROC) and
b) confusion matrix (CM) to estimate parameter's meters accuracy, roc source, precision.

4.4.1 Receiver Operation Curve (ROC)

ROC curves indicate a binary classifier system’s diagnostic capability when discrimination threshold is modified. It was created in 1941 by military radar receiver operators. A method was devised then [9]. The ROC curve is produced by comparing TPR and FPR at multiple threshold values. The true-positive rate is called sensitivity, recall, or detection. Multiplying specificity by 1.5 yields false-positive rate or alarm probability. Power and decision rule type I error may be graphed. Fall-out-dependent ROC curve sensitivity or recall. Plotting the cumulative distribution function (area under the probability distribution from to the discriminating threshold) of the detection probability on the y-axis and the false-alarm probability on the x-axis produces the ROC curve. This is possible using detection and false alarm probability distributions.[100]

4.4.2 The Confusion Matrix (CM)

An error matrix, or confusion matrix, is a contingency table used to explain a classifier's performance when the "truth" is known. In a confusion matrix, each column (or row) reports the predicted class, such as the number of predicted diseases or normal individuals, while each row reports the true class, such as the number of actually occurring diseases or truly normal individuals. Four numbers are given [101].

1. true positive (TP) measures the percentage of positives properly predicted given genuine positivity.
2. Second, false negative (FN) measures the proportion of projected negatives given actual positivity.
3. False positive (FP), 4) True negative.
4. True positive (TN) is the proportion of positives (or specificity, the fraction of expected negatives that are negative). A better classifier should be more sensitive and specific, Table 4.2 for confusion matrix idea.

Table 4.2 confusion matrix concept

		Predicted Condition	
		Disease	**Normal**
True Condition	**Disease**	**True Positive (TP)** **(Sensitivity)**	**False Negative (FN)**
	Normal	**False Positive (FP)**	**True Negative (TP)** **(Sensitivity**

4.5 The Strock Detection Results from EEG path

The algorithm types with result it is reviewed four types of algorithms

- Multi-layered perception (MLP)
- Decision Tree (D.T)

- Extra-Tree (E.T)
- Random forest (R.F**)**

The dataset in in this work import from the reference[69] [70]. The dataset comprises of forty (40) individuals who have a history of having an ischemic stroke and forty (40) persons who are healthy. Patients have *a mean age* of 72, with a *standard deviation* of 13.6, while healthy individuals have a *mean age* of 73, with a *standard deviation* of 7.1. EEG signals were captured at 256 Hz and recorded for periods ranging from 15 minutes to 4 hours for each participant.

In my direction project, data from four channels and two reference channels are collected, as shown in (Figure 3.6)., in that direction used a ROC technique as a good tools to analysis the signals and to collect the data ,as a *three parameters, or criteria*

a. Score
b. Recall
c. F1-score

As evaluation metrices as accuracy and ROC score are not sufficient for evaluation the model. ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

The true-positive rate is also known as *sensitivity*, recall or probability of detection in machine learning. Since ischemic stroke detection is a binary classification problem,

For a deeper understanding of how each algorithm performs, we've included the confusion matrix. The Keras [102] Python libraries serve as the foundation for our implementation. Alongside scikit-learn.

In that job there is two major case *patient group* denoted by(P), *healthy group* Denoted by (H), So, there is two classes P AND H, here after extract the feature now starting The operation with machine learning path and select four algorithms above. The operation stared to Selected randomly some as training signals and some as test signals so that the validation of our classification methods is carried out as follows:

(1) For each of the classes $C_{l,,L}$ = {0, 1}, containing Nl epochs of the patients or healthy person, randomly choose NlT matrix curves as the test set and the rest (Nl − NlT) as the training (library) set.

(2) Different Algorithms were applied on the features extracted from each epoch.

(3) The above steps are repeated Q times (Q-fold cross validation), each time choosing different sets of training and test feature in Cl. The probability of correct classification for each class can then be estimated by $\overline{\boldsymbol{p}}_{\mathbf{cl}} = \frac{\mathbf{1}}{\boldsymbol{Q}} \sum_{\boldsymbol{q}=\mathbf{1}}^{\boldsymbol{Q}} \overline{\boldsymbol{q}}_{\mathbf{c1q}}$

where $\overline{\boldsymbol{p}}_{\mathbf{clq}}$ denotes the estimated probability of correct classification of class l at the qth trial, q = 1, . . ., Q, i.e., $\overline{\boldsymbol{p}}_{\mathbf{clq}} = \frac{\boldsymbol{N_{Lc}}}{\boldsymbol{N_{lT}}}$ with $\boldsymbol{N}_{\mathrm{LC}}$ and $\boldsymbol{N}_{\mathrm{LT}}$ being the number of correct classification and total number of members in class C_l at the qth trial.

- For ensuring the accuracy of the finding forty (40) distinct permutations of test and training datasets were used in the education and examination of all classifiers. The data from the forty 40 healthy people and the 40 sick are randomly split up into a training dataset and a test dataset. The training set contains 80% of the data, both healthy and sick, while the test set contains 20% of the data. There is not a single patient that appears in both the training and the testing data. the algorithms performance shown Figure 4.3 below, can be seen it represent a result for the four algorithms

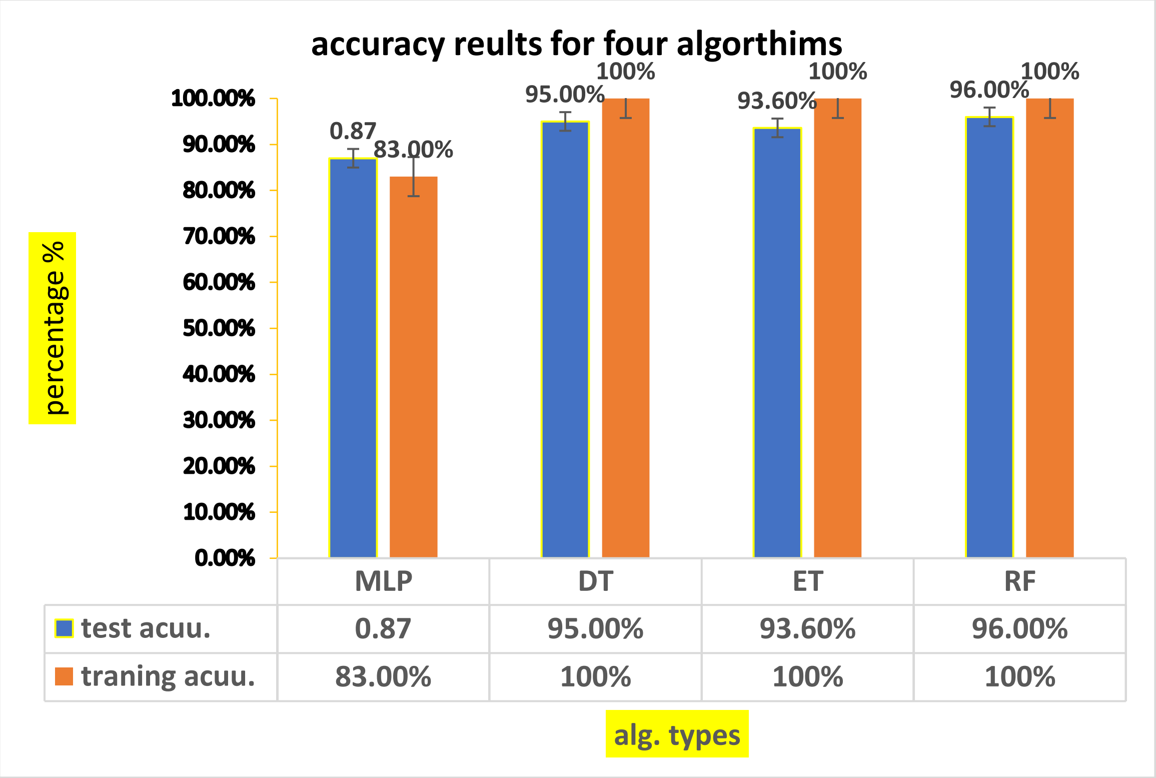


Figure 4.3 The algorithms performance accuracy

- The Blue refer to the accuracy of the test datasete
- The Orange represnat the accracy of the traning datesete

4.5.1 Accuracy:

As can be seen from figure 4.3, the performance of the bootstrap classifiers remains stable regardless of the Both the correctness of the training, which is 100%, and the accuracy of the exam, which is about 93% of the total. The accuracy of the MLP is much lower than that of the DT. and ET classifiers, but it still accomplishes a level of accuracy that is acceptable. It is noted that there has been no notable shift in the performance about the precision of MLP on both the training and the testing datasets.This suggests that the MLP model may not be a perfect match for the data. A further extensive architectural planning may be used.

4.5.2 The ROC Analysis with Results

ROC. In the context of the Receiver Operating Characteristic (ROC) score value approaching 1 (or 100%) is often seen as indicative of excellent performance. In general, the receiver operating characteristic (ROC) score of all three classifiers exceeds 80%. The mean receiver operating characteristic (ROC) score of these models exceeds 85%. The analysis of Figure 4.4 reveals that the micro-average ROC score of the MLP classifier surpasses the overall ROC curve score (shown by the black line). This discrepancy suggests that the dataset is imbalanced, thereby impacting the performance of the MLP classifier. The Random Forest model shown in Figure 4.7 has considerable promise and stability in comparison to the other two models.

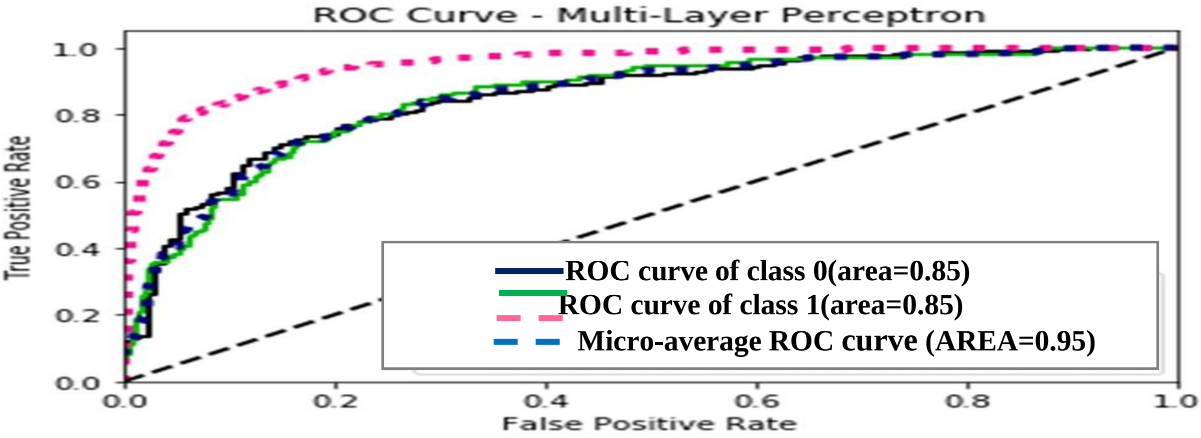


Figure 4.4 Multi-Layered Perception Roc Curve.

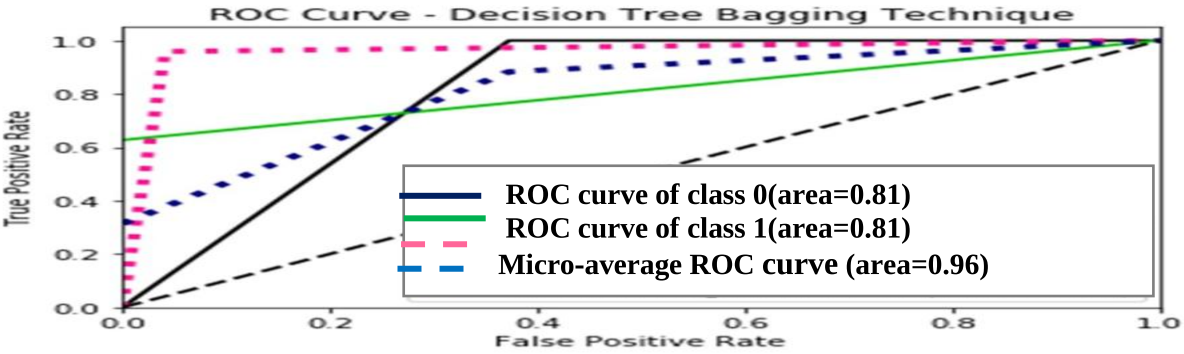


Figure 4.5 Decision Tree Roc Curve.

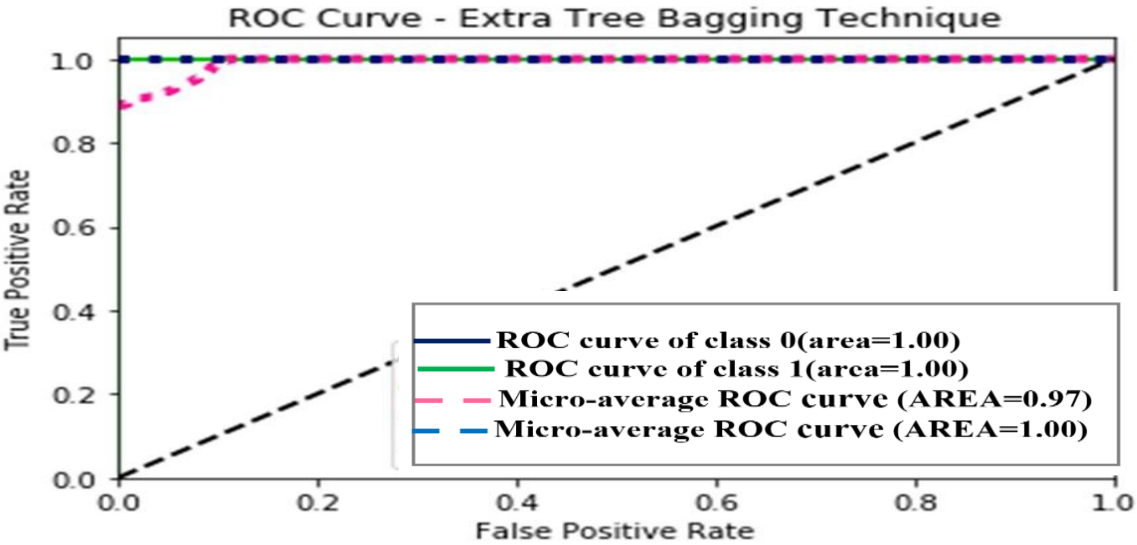


Figure 4.6 Extra-Tree Roc Curve.

The random forest model shown in (Figure 4.7) is the most promising and stable so far as compared to the other two models.

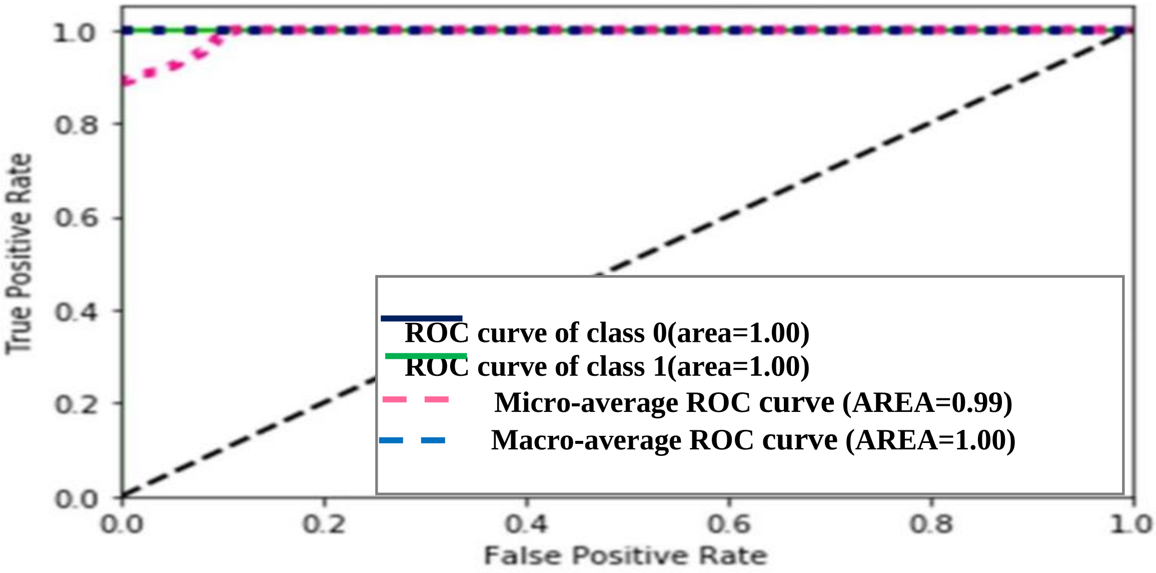


Figure 4.7 Random Forest Roc curve.

The random forest model which shown in (Figure 4.7) is the most promising and stable so far as compared to the other three models.

4.5.3 Confusion matrix analysis with results

The confusion matrix is a useful criterion for determining the outcomes, and its accuracy may be compared across a number of different algorithms. TPR and FNR are both provided by the confusion matrix. The ROC curve score and accuracy percentage are validated and verified as a result of this. The confusion matrices of the various models that are shown in Table 4.3

- The bootstrap models, when applied to both healthy and patient data, are able to accurately predict healthy datasets with a precision of one hundred percent (Table 4.3), although there is some fluctuation with test datasets (Table 4.4).

- more than eight times as many epochs are included in healthy datasets as are included in sick datasets, giving healthy datasets a distinct advantage. This results in an issue with the data being imbalanced.it can be show in Table 4.3
- The Traning Datasete results

Table 4.3 Confusion matrix of all four models' performance withTraining datasets

Training dataset		
MLP	**H**	**P**
H	**2500**	**2150**
P	**860**	**1100**

Training Dataset		
D.T	**H**	**P**
H	**4650**	**0**
P	**0**	**1300**

Training Dataset		
E. T	**H**	**P**
H	**4650**	**0**
P	**0**	**1300**

Training Datasets		
R. F	**H**	**P**
H	**4650**	**0**
P	**860**	**1300**

where

MLP --- refer to Multilayer participants

D.T Refer to Decision Tree

E.T Refer to Extra Tree

R.F Refer to Random Forst

P the paint participant

H the healthy participant

The Test Datasets in table 4.4

Table 4.4 Confusion matrix of all four models' performance with the test dataset

Test dataset		
MLP	**H**	**P**
H	**770**	**50**
P	**90**	**60**

Test dataset		
D.T	**H**	**P**
H	**820**	**0**
P	**52**	**98**

Test dataset		
E. T	**H**	**P**
H	**820**	**0**
P	**49**	**101**

Test dataset		
R.F	**H**	**P**
H	**820**	**0**
P	**35**	**115**

It is possible to draw the conclusion that the data imbalance has an effect on the overall performance of all models. For instance, the healthy dataset (Figure 4.8) has a total number of 4650 healthy datasets, but the patient dataset only contains 1300 datasets, indicating that the healthy dataset is much larger in terms of its overall number of participants. E.T can manage datasets with an imbalance somewhat better than MLP and DT and random forest is the best.

4.5.4 classification results

Detailed Report on Classification, when compared to MLP, D.T and E.T, R.F both have a greater overall accuracy, recall, and F1-score than 90% percent, which indicates that these three classifiers have accurately predicted outcomes. Despite this, the MLP accuracy, recall, and F1-score are all within acceptable ranges.(Figure 4.8).

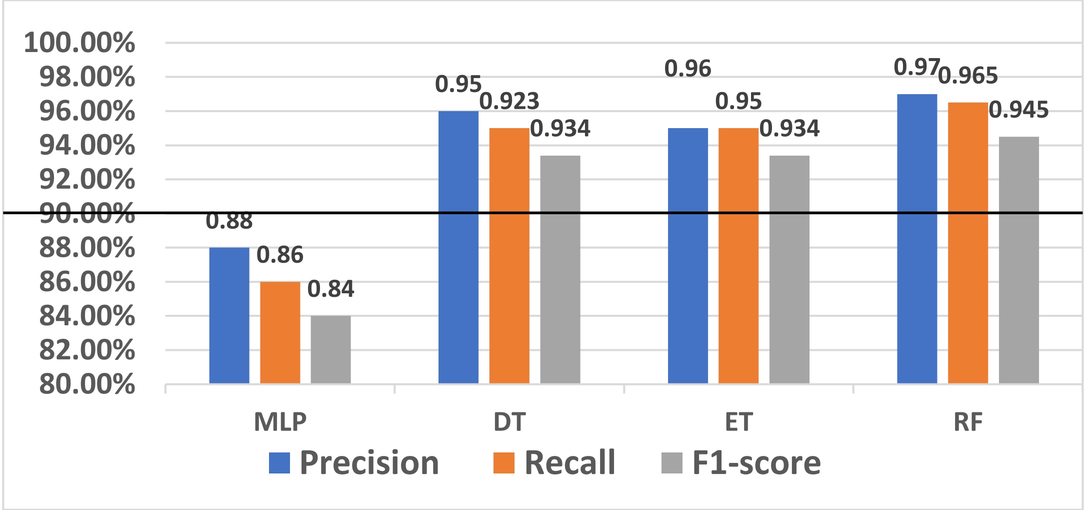


Figure 4.8 Average Precision Recall and F1-Score of all classifiers. The black line marks 90% benchmark.

When there is no problem of data imbalance, it is common to use machine learning algorithms such as Multi-layered Perceptron, Decision-Tree, Extra-Tree. Random fprset The performance of all classifiers is evaluated via the use of three distinct performance metrics.

4.6 The Image Processing (IP) results

Refer to the chapter 3.9 section the genral block diagrm that refer to image processing (IP) is shown in this part which shows more detailes about the (IP) exacley with CNN path now it is more advance to get the results her suggest the preposed path to follow the steps, [103].

The proposed algorithm can detect the brain strokes in early stage with enhanced time complexity using brain MRI images. Proposed algorithm consists of various steps like Image Acquisition, Pr-processing, Segmentation, Binary image conversion, Feature extraction and decision-classification using machine

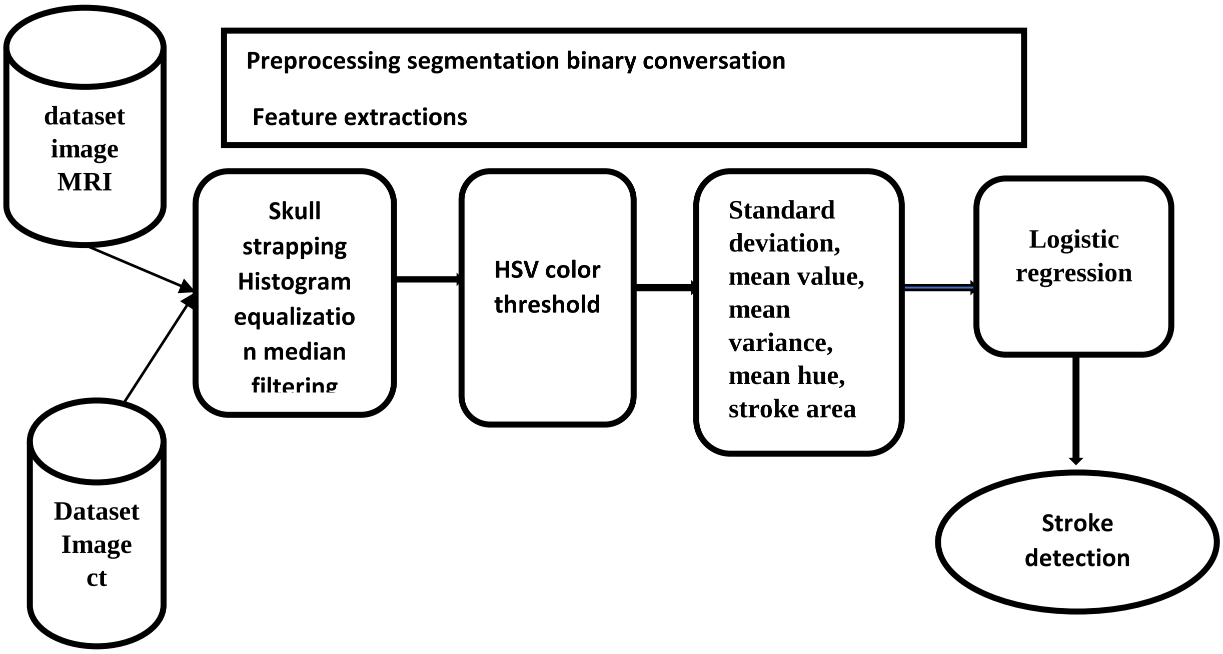

Preprocessing segmentation binary conversation
Feature extractions
dataset image MRI
Dataset Image ct
Skull strapping Histogram equalization median filtering
HSV color threshold
Standard deviation, mean value, mean variance, mean hue, stroke area
Logistic regression
Stroke detection

Figure 4.9 illustrates the proposed model with IP path

Learning model. Figure 4.9 illustrates the proposed model. The proposed algorithm can detect the brain strokes in early stage with enhanced time complexity using brain MRI images. Proposed algorithm consists of various steps like Image Acquisition, Pr-processing, Segmentation, Binary image conversion, Feature extraction and decision-classification using machine learning models.

4.6.1 Image Acquisition

Machine learning begins with data gathering or image acquisition. This stage collects data from many sources. picture acquisition captures and stores an object's picture in a database for processing. Brain pictures are usually acquired using CT and MRI. CT scans employ X-rays to capture body pictures, whereas MRIs use radio waves and magnetic fields to create brain images. MRI is more sensitive than CT in detecting brain stroke because it can detect small changes in brain tissues. Hence This study extracts features from MRI images [95]

4.6.2 Image Pr-processing

Digital image processing pre-processes MRI pictures. To improve feature extraction, skull stripping, histogram equalization, median filtering, and grey image conversion are done.

- Skull stripping: Brain MRI scans usually show fat, skin, muscles, eyeballs, and neck. This non-brain tissue hinders automated brain analysis and picture segmentation. Thus, quantitative brain MRI morphometric research usually needs a preliminary picture. pre-processing to isolate the brain from extra-cranial or non-brain tissues. Skull stripping is the process of isolating brain tissue from non-brain tissue from an MRI image of a brain. In this research intensity-based method is used for Skull stripping
- Histogram Equalization: Low-contrast MRI pictures make diagnosis harder. To enhance this, localize pixels better. Histogram Equalization improves picture quality. The brightness and contrast are improved to retain brightness and information.
- Median Filtering: Medical research requires maximum noise removal before surgery. Median filtering removes high-frequency noise without distorting picture edges. The median of adjacent pixels determines freshly denoised
- pixels in median filtering. As the newly computed pixel is based on nearby pixels, it reduces noise while keeping picture quality.

4.6.3 Segmentation

Segmentation selects intriguing items or crucial areas from a picture. Segmentation divides an image into mutually exclusive zones that are spatially contiguous and contain only homogenous pixels depending on specified criteria. The suggested approach segments brain portions on MRI gray scale images using

HSV (Hue, Saturation, and Value) color threshold. It eliminates extraneous brain MRI gray scale pictures outside the color spectrum. In this study, segmentation removed the non-stroke region of MRI picture and showed the stroke. Hue, brightness, and saturation determine HSV color thresholding segmentation. Figure and 4.10(b) demonstrates MRI brain segmentation with HSV threshold. Figure 4.10(a) illustrates the image conversion after the segmentation on the image which contains the brain stroke, while Figure 4.10b) shows the image segmentation for brain MRI images which does not contain brain stroke

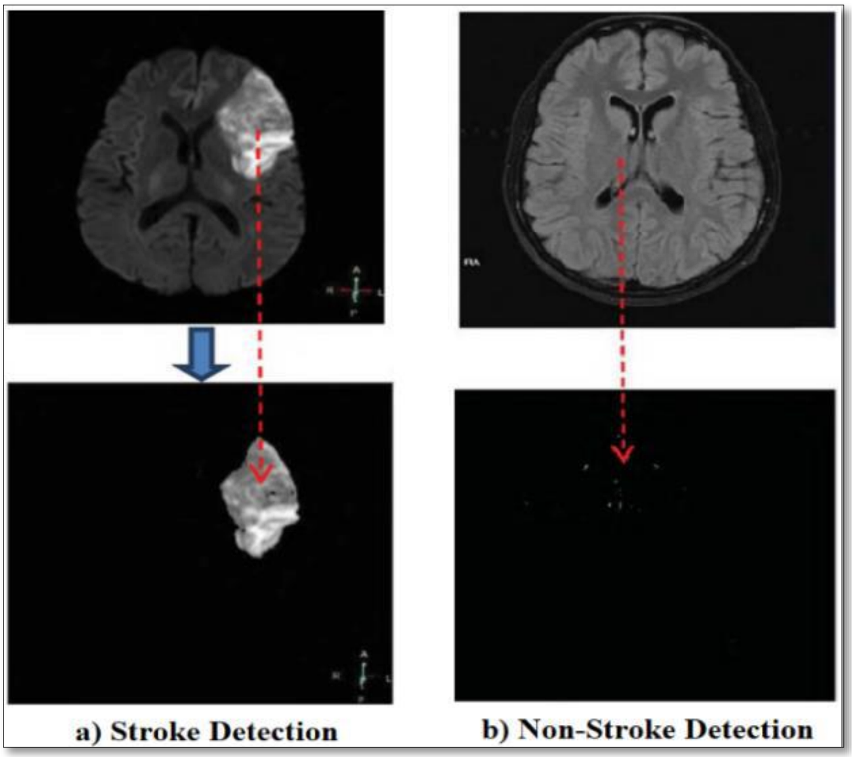

a) Stroke Detection b) Non-Stroke Detection

Figure 4.10 segmentation on the MRI brain using HSV threshold technique

4.6.4 Binary Image conversion and feature extraction

Next, segmented brain picture is entered and binary image conversion is used to create a binary image. Binary pictures can only have black and white pixels. White pixels are 1 and dark pixels are 0. This decreases computing needs and increases algorithm time complexity.

Important features and characteristics are picked during feature extraction to improve machine learning accuracy. Since actual data may include numerous features, employing all of them in training may impair accuracy, therefore only useful and non-redundant features may be extracted. Standard deviation, mean variance, mean hue, and brain stroke area are utilized to train machine learning models in this study. Mean hue is the color pixel's dominating wavelength of stroke component, separating undesirable from stroke portion.

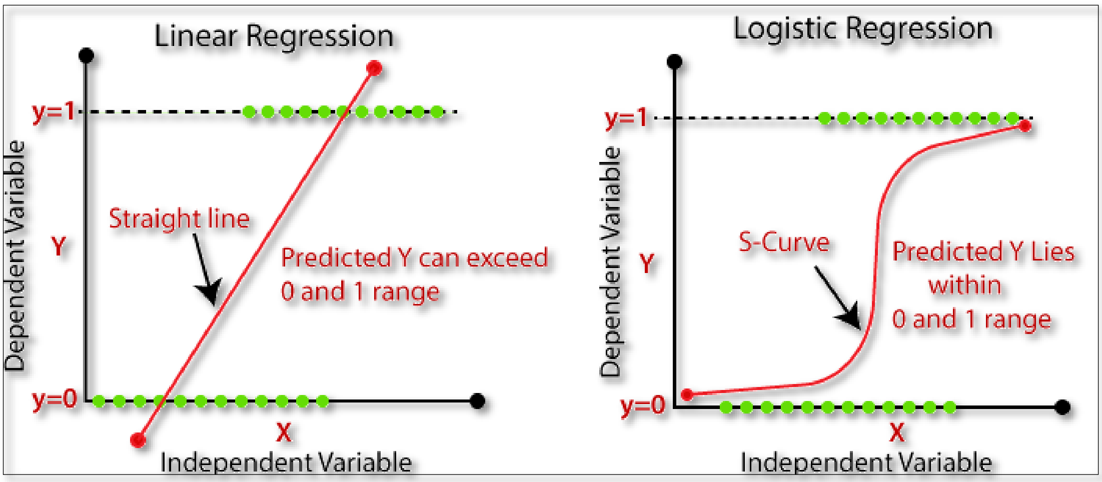
- *Standard deviation* measured pixel dispersion. The brain stroke detection accuracy improves when pixels have less fluctuation and are near the brain stroke part of an image.
- *Variance measures* data point deviation from the mean. It compares average pixel value to random pixels.
- *Affected Area*: Another essential stroke detecting feature. Stroke intensity depends on stroke size. Mini strokes and ischemic strokes vary in size.

4.6.5 Logistic Regression Classifier

Logistic Regression Classifier determines independent-dependent relationships. This classifier is typically employed in categorical classification systems with fixed outcomes, making it a discriminative model. Supervised machine learning algorithm Logistic Regression Classifier. Logistic regression reduces data overfitting. Thus, model accuracy is comparable to sample data vs.

Real-time data analysis.Regression analysis models variables to predict a variable based on its relationship with one or more independent variables[103]

The predictors or independent variables are used to predict the criteria or dependent



(a) (b)

Figure 4.11 Logistic and Linear regression model

variable. Dichotomous variable prediction is logistic regression over linear regression. This behavior is largely employed in medical machine learning studies to identify illness. Figure 4.11 shows the difference between modeling of dependent variables using linear regression Figure 4.11(a) and logistic regression Figure 4.11(b). Logistic regression helps to determine the value strictly between 0 to 1, no matters where locate on the x axis [103] .

$$f(z) = \frac{1}{1+e^{-z}} \qquad (4.1)$$

The logistic function is in Equation 4.1 This equation uses e-z as the logistic function and f(z) as the logistic regression output. The logistic function in linear regression represent in equation 4.2.

$$Y = b1.x1 + b2.x2 + \cdots\cdots + bk.xk + a \qquad (4.2)$$

Y is the dependent variable, x is the independent variable, a and b are regression coefficients.With this equation, logistic regression may be rewired as,in equsion 4.3

$$f(z) = \frac{1}{1+e^{-(b1x1+b2x2+\cdots\cdots+bkxk+a)}} \qquad (4.3)$$

In this work, python sklearn library is used to perform *logistic* regression. Code block used for *linear* regression shows in Figure 4.12 below:

The simple software code in python language attachment in appendix -D under the title source code program lin.py, refer to linear regression, and the source program log.py, refer to logistic regression

The output of the program lin.py

```
the data x [5, 7, 8, 7, 2, 17, 2, 9, 4, 11, 12, 9, 6]

the data y [99, 86, 87, 88, 111, 86, 103, 87, 94, 78, 77, 85, 86]
```

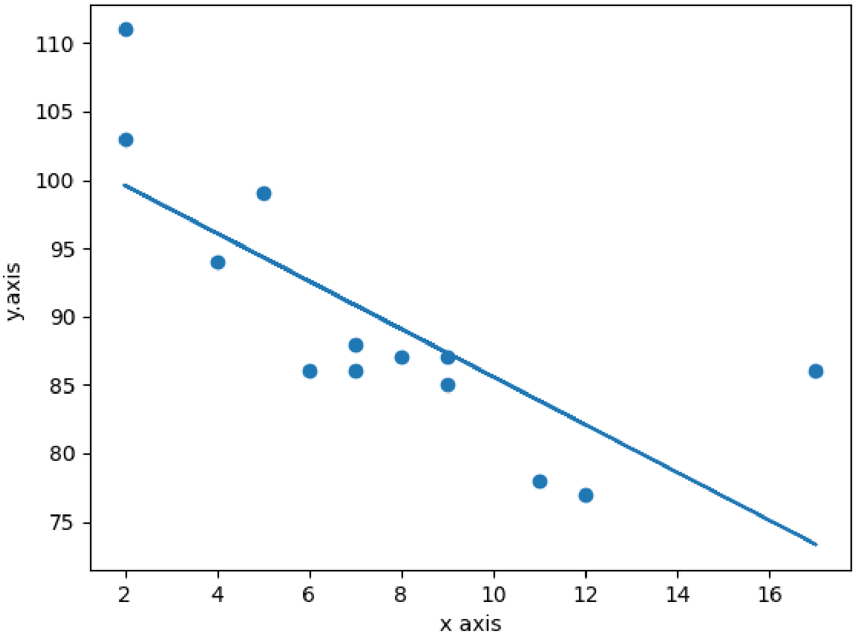


Figure 4.12 Linear regression model

The program log.py is listed in appandix D the output to that program shown in in figur 4.13 below

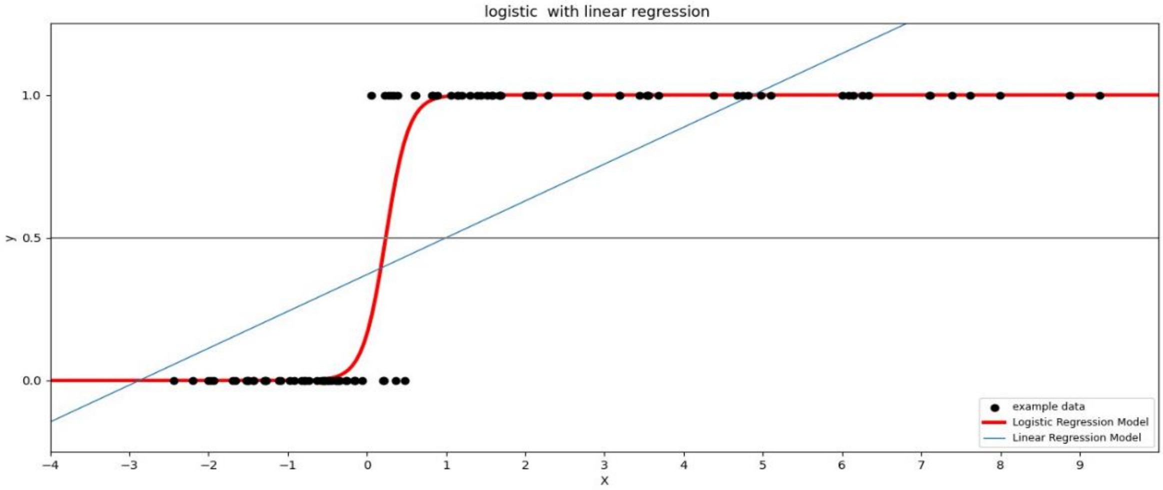


Figure 4.13 logistc regression ouput program(log.py)

4.7 Result Analysis

This section focuses on the Dataset collection, system configuration and performance analysis of the proposed system using comparison of accuracy rate.

The dataset was collected from the al-hilla hospital ,and al imam al sadiq hospital [90]

4.7.1 Dataset and System Configuration

The study uses 1000 MRI pictures for outcome analysis. This data is split for training and testing.JPG photos with varied file sizes are utilized in this investigation. Microsoft Windows with 16GB RAM. Python ver 3, pycharm is used for data preparation and sklearn for machine learning model training.

4.7.2Analyses Based on Experiments

The results that were produced from the suggested study are compared with the current method of brain stroke detection for the purpose of experimental analysis. After gathering pictures of the brain, scientists used the data to several already-developed algorithms, such as the K-nearest neighbor method, the Random Forest ensemble algorithm [104], and linear regression. For the purpose of the experimental investigation, one thousand samples will be evaluated. The performance of the K-nearest neighbor approach, the Random Forest ensemble technique, linear regression, and the suggested algorithm are compared in Figure 4.14, which analyzes the correct results derived from the analysis of 1000 data samples. When doing analysis on the dataset, splits of 90-10 are used, with 90% of the dataset being utilized to train the model and the remaining 10% being applied to test the dataset.

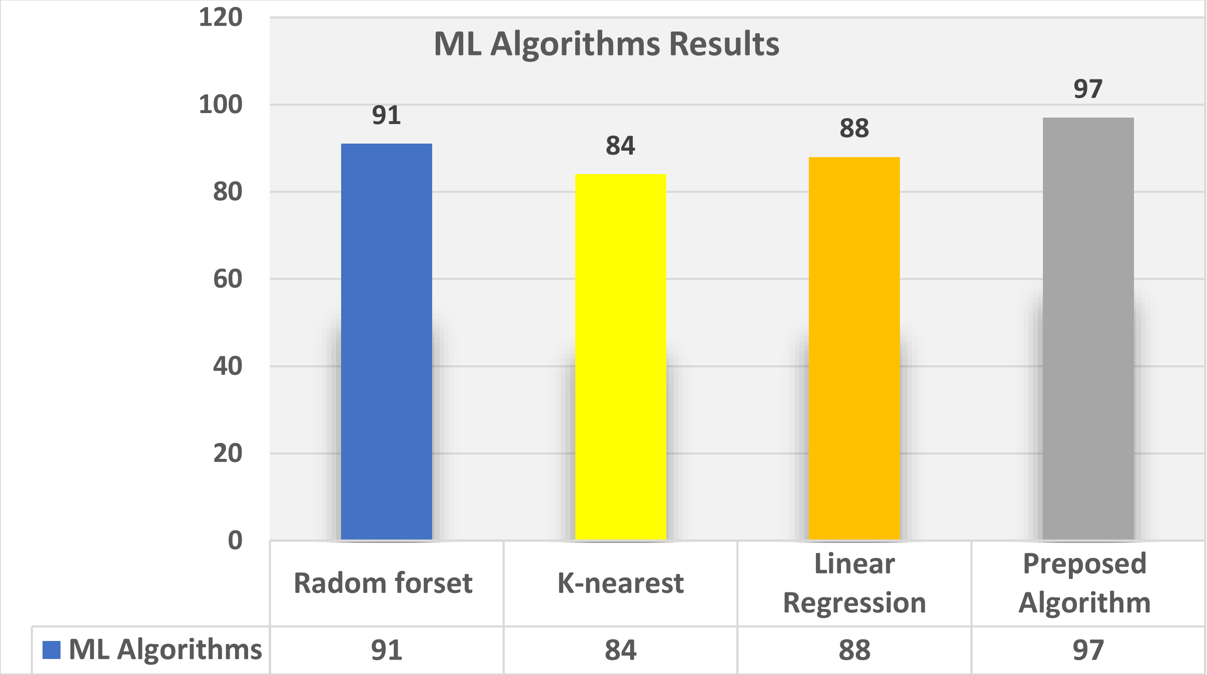


Figure 4.14 show the results among all algorithms

Figure 4.14 demonstrates that the proposed approach has better accuracy for detecting brain strokes than the K-nearest neighbor strategy, the Random Forest ensemble technique, and the Linear regression algorithm. K-nearest neighbor and Random Forest, two popular machine learning classifiers, had the lowest accuracy, followed closely by Linear regression.

In lieu of the split test, new data must be supplied to assess the algorithm's performance in a real-time context. One thousand freshly collected brain MRI images that were not included in the training set are used in this assessment.

The results of the exam were shown in Table 4.5. As Table 4.5 shows, the accuracy of all algorithms goes down when unknown real-time data is used to test them. K-nearest neighbor and linear regression became much less accurate. But, because When testing, the ensemble method is used by random forest. Its accuracy with real-time, new data is about the same as that of the split test. But the precision of Logistic regression didn't change by much. This shows that the proposed method works well both with split tests and with datasets that aren't known.

Table 4.5 Real time result comparison

Algorithm	Total Instances	Correctly Classified	Accuracy Percentage
K-nearest neighbor	**1000**	**846**	**84.6%**
Random Forest	**1000**	**913**	**91.3%**
Linear regression	**1000**	**880**	**88%**
Proposed Algorithm	**1000**	**981**	**98.1**

4.8 some conclusion in that path

A brain stroke happens in the human body either when the blood vessels that are present in the brain rupture or when the blood flow to the brain is obstructed. Strokes to the brain provide a significant risk of either death or significant disability. Patients may be saved from additional injury if they are diagnosed with a brain stroke or restricted blood supply to the brain as soon as possible.

The purpose of this work was to offer an effective method for the diagnosis of brain strokes utilizing image processing and logistic regression. There are several approaches that may be used to identify a brain stroke, and researchers are working hard to modify each of these methods so that they can detect a stroke in its earliest stages.

Image acquisition, preprocessing, segmentation, binary image conversion, feature extraction, and logistic regression are all steps that must be taken in order to reach the desired outcome. In conclusion, the results of the suggested method are evaluated and contrasted with those of an artificial neural network and an algorithm for machine learning. The suggested approach achieved an accuracy of 99.2% when applied to split test data and 98.1% when applied to real time unknown data, respectively. It was discovered that the accuracy of the suggested method for brain stroke diagnosis is better than that of the K-nearest neighbor approach, the Random Forest ensemble technique, and the Linear regression algorithm that are already in use.

4.9 Image Processing Implementation with Results

From the block diagram Figure 4.9 above, all steps can be implemented by the python program with flowchart Figure 4.15 that content all above steps as follows

- Normalization
- Filtration
    - ✓ Averaging
    - ✓ Grayscale
- Segmentation
    - ✓ Contain the Otsu thresholding type.
- Morphology

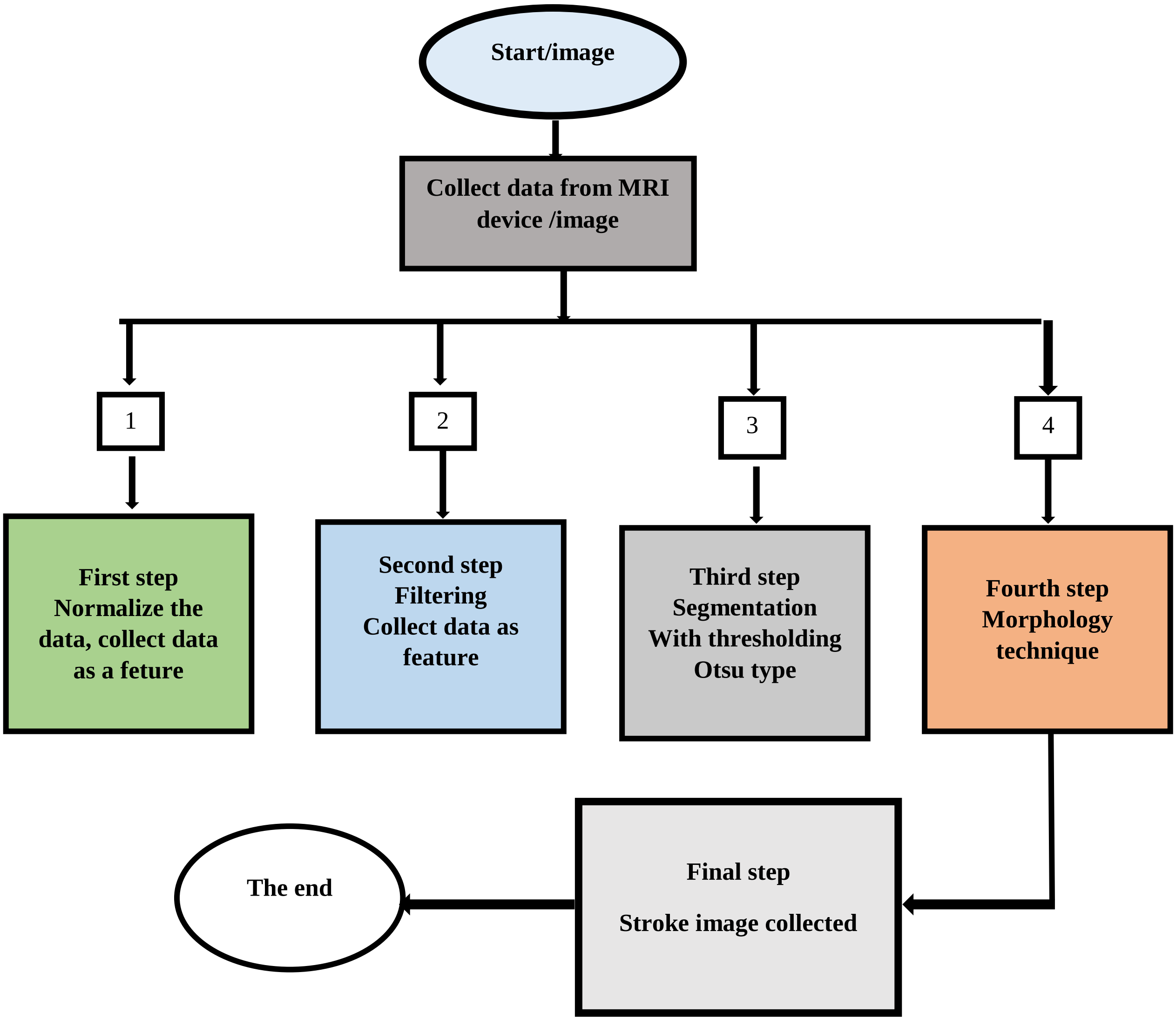

Start/image
Collect data from MRI device /image
1
2
3
4
First step Normalize the data, collect data as a feture
Second step Filtering Collect data as feature
Third step Segmentation With thresholding Otsu type
Fourth step Morphology technique
Final step
Stroke image collected
The end

Figure 4.15 the flowchart operation in the image processing path *1,2,3,4 refer to the subroutine to each task*

- The source program list in the Appendix E (yyy-11.py) written with python program ver 3.0, and satisfy the above block diagram In Figure 4.15, The output for each steps of images as follows, Figure 4.16 when the program is executed (yyy-11.). program plot four case of image types refer to the source program yyy-11.py ***1-orignal image 2- grayscale image 3-thresold image 4-segmatation image.***

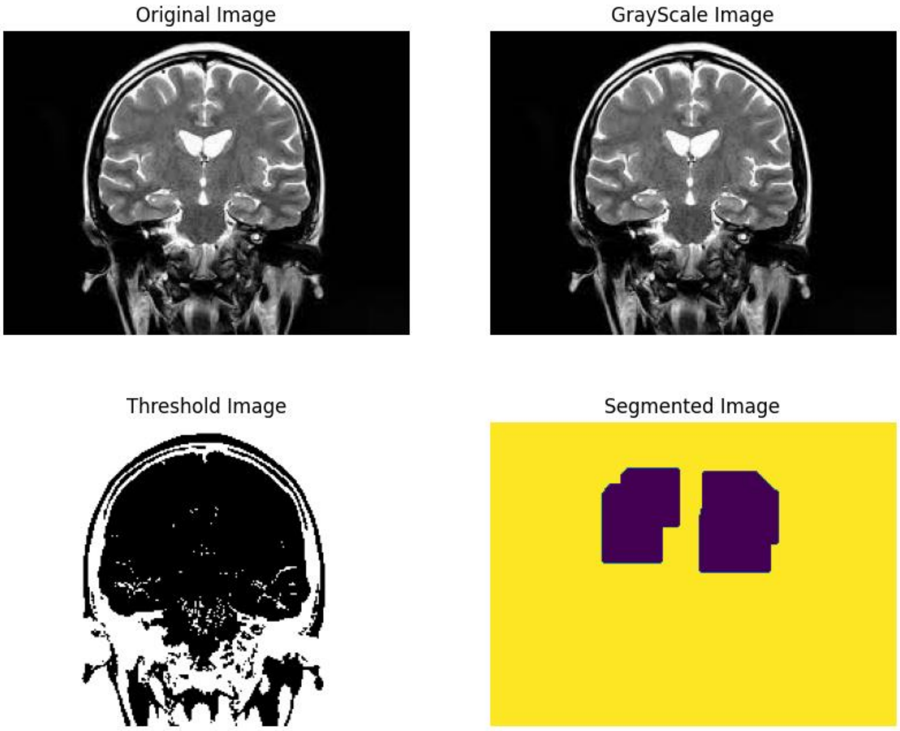


Figure 4.16 show the image processing steps

- Illustrated the source program to the morphology technique it’s one of the important steps to enhancement the view the image with Morplogy-1.py that program attachment the source list in Appendix -E, Figure 4.17 show the transform from original image to morphology image do in that program.

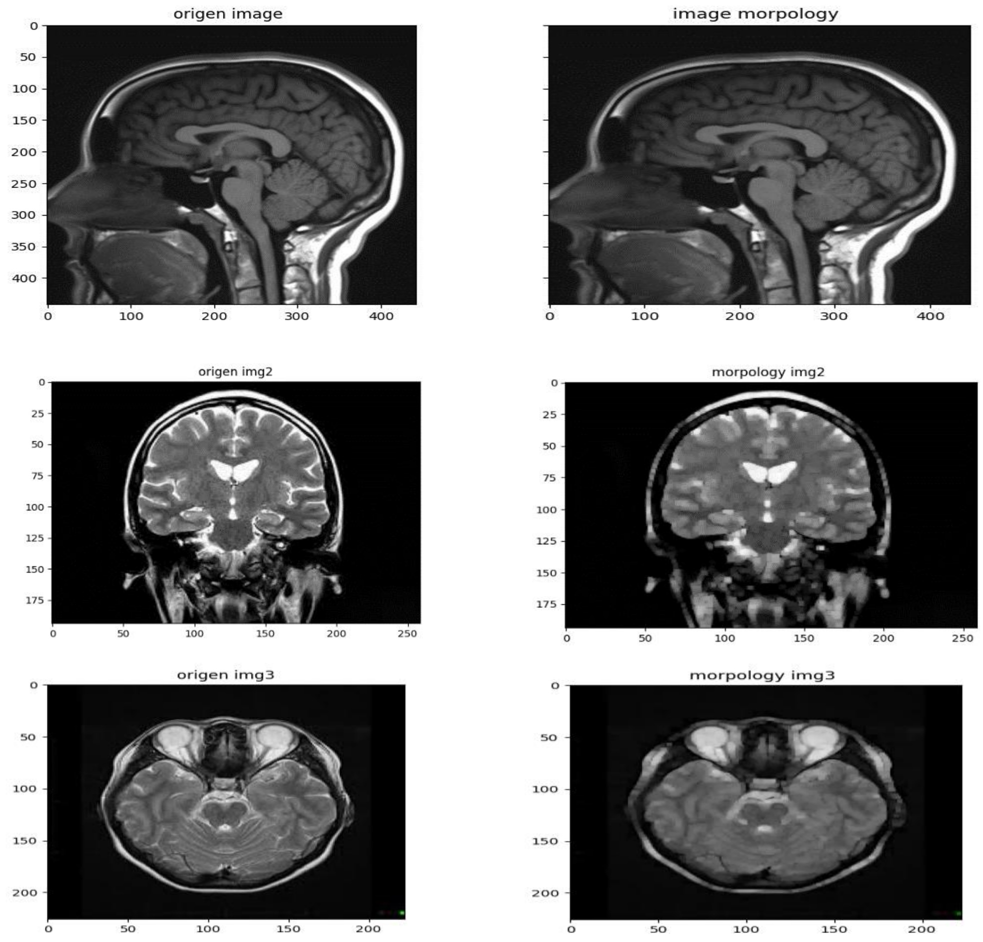


Figure 4.17 show the original image with morphology for three images

The Appendix -E refers to the feature extraction to the image processing path that represent in the above flowchart Figure 4.15, source program(feature-77.py).

With some result. In the Appendix -E, collect case of the stroke from one participant and do the processing that remember in figure 4.9 (proposed method) in the image processing path, the display of image contain thirty plot in that Appendix -E, display some plots to six attributes of the image.

1. Original image
2. Normalized image
3. Grayscale image
4. Threshold image

5. Averaging image
6. Morphology image

And collect all to get thirty (30) items that show in Appendix -E

In Appendix F illustrated the data from images for three sample of image, after resize the original data image 256x256x3 to decreasing the dimension to 9x9x3 in order to display and backup here in the work, because the data is huge and cannot backup here.

In that Appendix F, list the python source program imdata.py to extract the data and display 27 table each table contain 27 item (data), all data is 486 item refer two into two image only, so the total of item (729)

4.10 Compare Between Two Paths

In this work explain two paths to deal with the stroke as a main target to detect and classification,

- first the EEG path
- the image processing path

, Tabe 4.6 show some information to compare between two path.

Table 4.6 Compare Between EEG Path and IP path

	The EEG path	**The IP path**
1	**Its more complex to extract the date – time series approach**	**It's easy for extract the data FROM The devices**
2	**The development with the fixed stroke it's not accurate**	**Its more advance to fix the stroke diseases**
3	**That field maybe active with another Disease does not stroke**	**That field have new more active device to can be fixed the stroke very accurate**
4	**In that field it cannot determine the stroke happened or not but can be show the symptoms as physically by a doctor**	**In that field can be exactly fix the case of the stroke**
5	**In that path there is new device that can be used but the doctor may be need more information exactly clinical about the patent to get good final decision**	**Clinical information to the patient gives good decision but it's not critical to obtain the final decision**
6	**The new technology in that field is available I illustrated some device in chapter 5, it can be get more good results to collect the data**	**Today that path its vey development to possibility to collect data by new more advance technology like pet scan that can be get more information to fix the stroke and expected the time exactly**
7	**He mistakes to fix the diagnostic line can be very predictable because, must be get a lot do information to help the doctors to fix the stroke or not**	**The diagnostic line her depend about the device output resolution and the efficiency of the doctors**
8	**IN the direction the feature extraction is more complex and this case because the data collect from multiple sensors and must be clean, and fix few numbers to possibility to control the data**	**In that direction the data collect as an image and can be easily to handle with and to extract the main point Easley**

Chapter five

Conclusion and Future Works

Chapter Five

Conclusion and Future Work suggestion

5.1 Introduction

The fundamental purpose of this research is to focus on the stroke case, how it happened, to attempt to anticipate the approximate time that may occur, and to choose new techniques to fulfil and seek to overcome this case in order to protect the life of any person. This study will be conducted in order to achieve these objectives.

There are innovative approaches that may be used right now, and these solutions have the potential to be regarded as exceptional work in the years to come. The time series route, which is a reference to the EEG, and the IP path line are the two key lines that are used in this kind of investigation.

Within the two methods that I stated before, there is further cutting-edge technology for electroencephalogram (EEG) devices (time series) and image processing (IP). In order to collect dates, several methods may be used, which will ultimately result in more precise conclusions.

5.2 Conclusion:

In this work select two paths to reach the goal (stroke detection)

a. Electroencephalogram (EEG) path +Machin leering (ML)
b. Image processing (IP) path + machine learning (ML)

Obtain some conclusion in each path.

5.2.1About the EEG path

1. Its non-invasive techniques so it’s more safety and no side effect.

2. It's more complex when extracting the feature attributes from the signals.
3. Because this approach represents a time series, it requires more time when capturing the information from any participants. This consideration is considered a negative because it must take a considerable amount of time.
4. It must given many sensors to snap up the stroke case and do more investigation.
5. While an electroencephalogram (EEG) may not always be the best tool for diagnosing and treating a stroke, it is a solid option for studying epilepsy.
6. A doctor's ability in determining the specifics of a stroke case and the advancements in the technology used in your room define the EEG route, so various doctors may come to different decisions on how to treat a stroke case.

5.2.2 About the image processing (IP)

about this path, concluded some points to fix and detect the stroke disease

1. Image processing represent the image path to collect the data from any participants, and following the processing in that path is more facility with more easy steps that can convert all formula to the steps in the MATLAB or Python-MNE program.
2. In that path IP can be used for more enhancement and filtering with morphology processing to get good results easily
3. Can almost get good results to determine the stroke with more accuracy, and fix the location and size.
4. The fast result can be obtained to fix the stroke and the doctor can decide stroke or not.
5. There are many varieties of devices like MRI, CT SCAN available that can be used to detecting the stroke, with position.

6. Many algorithms can be used with the IP refer to machine learning and deep learning and obtain more accuracy.
7. With stroke disease it can be fixed that the IP choice is the best selection

5.2.3 Machine Learning (ML) field

In that way, it depends on

1. An amount of dataset (size) is a very important point, if can be obtained huge of the data for a group of participants, in this case, machine learning is the preferable choice
2. In that field ML, can select more algorithms that depend on the type of disease like stroke, epilepsy, and Parkinson's disease, also depend on what is needed, classification, detection, and prediction, that many algorithms operated under some condition.
3. Automatic learning machines ML obtained the findings of the study by using two different methods, and it is abundantly evident that the IP method is superior to the other one.

5.3 Future suggestion with Stroke Disease

The work that will be done in the future to detect or predict strokes is going to be extensive; the creation of the device and the algorithms represents the important points that can get the best results; however, strokes are still extremely hazardous to one's life, and there is a pressing need for additional research to develop an early warning alarm that will indicate any strokes in order to prevent them from occurring.

5.3.1 Progress on the EEG Protocol for Stroke Patients in the Future

In that path, future work refers to the development of new devices that can be located more accurately and can record new types of formats of datasets these data are used easily with the python-mne, and eeglab programs and get the best result, these device like

1. Bittium NeurOne™ Tesla EEG System: Unlocking Brain Secrets the Bittium NeurOne™ is a fast and precise electroencephalography (EEG) equipment for clinical and research usage. It is designed for medical professionals and scientists to detect high accuracy.
2. Nihon Kohden's EEG-1260 is a flexible and complete recording system that can be used for EEG, long-term epilepsy, cEEG ICU tracking, and sleep tests. The signals that come out of Nihon Kohden amps are known to be very good. For many years, our amplifier engineers have worked on making them last, being creative, and processing signals. One of our amps has 32 channels, and our newest ones have 64 to 256 channels in a thick grid.

5.3.2 Future suggestion in Image Processing (IP)With Stroke Disease

The path that involves image processing is essential to the progression of the task and may result in the creation of new branches that provide higher dependability. Additional research and the development of new algorithms are two of the many methods that may be used to obtain datasets from a variety of different devices.

Selecting various methods may boost work.

Can pick these image processing line techniques.

1. Magnetic resonance imaging (MRI)
2. A computerized tomography (CT)
3. Positron emission tomography (PET/CT)
4. Functional magnetic resonance imaging (fMRI)
5. Single photon emission computed tomography (SPECT)

The most important objective is to identify and categories the stroke, as well as to anticipate the occurrence of the event. Data collection from the patient may be accomplished by any of the aforementioned methods.

Increasing the amount of information that is collected from the patient is regarded to be the winning point. All of the techniques described above may be used as additional pathways to gather the data, which allows for the extraction of many new characteristics and the improvement of the accuracy of the findings collected.

References

References

[1] G. Foucher and S. Faure, "What is a stroke?," *Actual. Pharm.*, vol. 59, no. 600, pp. 57–60, 2020, doi: 10.1016/j.actpha.2020.09.020.

[2] S. Lek and Y. S. Park, "Multilayer Perceptron," *Encycl. Ecol. Five-Volume Set*, pp. 2455–2462, 2008, doi: 10.1016/B978-008045405-4.00162-2.

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

[4] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Control. signals Syst.*, vol. 2, no. 4, pp. 303–314, 1989.

[5] W. Vogt, "Decision Tree," *Dict. Stat. Methodol.*, 2015, doi: 10.4135/9781412983907.n503.

[6] S. Madeh Piryonesi and T. E. El-Diraby, "Using machine learning to examine impact of type of performance indicator on flexible pavement deterioration modeling," *J. Infrastruct. Syst.*, vol. 27, no. 2, p. 4021005, 2021.

[7] C. A. B. de Andrade, "Automatic Malware Analysis Based on Machine Learning: a Comparative Study of the WEKA and Rapid Miner Tools".

[8] R. Shukla, M. Gyanchandani, R. Sahu, and P. Jain, "Analytical Approach to Genetics of Cancer Therapeutics through Machine Learning," in *Soft Computing Applications and Techniques in Healthcare*, CRC Press, 2020, pp. 1–10.

[9] Y. Amit and D. Geman, "Shape quantization and recognition with randomized trees," *Neural Comput.*, vol. 9, no. 7, pp. 1545–1588, 1997.

[10] J. N. Fink *et al.*, "The stroke patient who woke up: clinical and

radiological features, including diffusion and perfusion MRI,” *Stroke*, vol. 33, no. 4, pp. 988–993, 2002.

[11] R. C. Hwa and T. C. Ferree, “Stroke detection based on the scaling properties of human EEG,” *Phys. A Stat. Mech. its Appl.*, vol. 338, no. 1–2, pp. 246–254, 2004.

[12] R. S. Jeena and S. Kumar, “A comparative analysis of MRI and CT brain images for stroke diagnosis,” in *2013 annual international conference on emerging research areas and 2013 international conference on microelectronics, communications and renewable energy*, 2013, pp. 1–5.

[13] B. J. Kim *et al.*, “Magnetic resonance imaging in acute ischemic stroke treatment,” *J. stroke*, vol. 16, no. 3, pp. 131–145, 2014.

[14] A. S. Al-Fahoum and A. A. Al-Fraihat, “Methods of EEG signal features extraction using linear analysis in frequency and time-frequency domains,” *Int. Sch. Res. Not.*, vol. 2014, 2014.

[15] S. K. Wijaya, C. Badri, J. Misbach, T. P. Soemardi, and V. Sutanno, “Electroencephalography (EEG) for detecting acute ischemic stroke,” in *2015 4th International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME)*, 2015, pp. 42–48.

[16] H. J. Audebert and J. B. Fiebach, “Brain imaging in acute ischemic stroke—MRI or CT?,” *Curr. Neurol. Neurosci. Rep.*, vol. 15, pp. 1–6, 2015.

[17] W. R. W. Omar, Z. Mohamad, M. N. Taib, and R. Jailani, “An analysis of EEG signal generated from ischemic stroke patient,” in *2015 Innovation & Commercialization of Medical Electronic Technology Conference (ICMET)*, 2015, pp. 74–77.

[18] J. Wu, R. Srinivasan, E. Burke Quinlan, A. Solodkin, S. L. Small, and S. C. Cramer, “Utility of EEG measures of brain function in patients with acute stroke,” *J. Neurophysiol.*, vol. 115, no. 5, pp. 2399–2405, 2016.

[19] P. Sivakumar and P. Ganeshkumar, “An efficient automated methodology for detecting and segmenting the ischemic stroke in brain MRI images,” *Int. J. Imaging Syst. Technol.*, vol. 27, no. 3, pp. 265–272, 2017.

[20] P. Vilela and H. A. Rowley, “Brain ischemia: CT and MRI techniques in acute ischemic stroke,” *Eur. J. Radiol.*, vol. 96, pp. 162–172, 2017.

[21] A. Subudhi, S. Sahoo, P. Biswal, and S. Sabut, “Segmentation and classification of ischemic stroke using optimized features in brain MRI,” *Biomed. Eng. Appl. Basis Commun.*, vol. 30, no. 03, p. 1850011, 2018.

[22] A. A. Qureshi, C. Zhang, R. Zheng, and A. Elmeligi, “Ischemic stroke detection using EEG signals.,” in *CASCON*, 2018, pp. 301–308.

[23] P. Garg, P. Kumar, K. Shakya, D. Khurana, and S. R. Chowdhury, “Detection of brain stroke using electroencephalography (eeg),” in *2019 13th International Conference on Sensing Technology (ICST)*, 2019, pp. 1–6.

[24] L. Shreve *et al.*, “Electroencephalography measures are useful for identifying large acute ischemic stroke in the emergency department,” *J. stroke Cerebrovasc. Dis.*, vol. 28, no. 8, pp. 2280–2286, 2019.

[25] S. Bhattacharjee, S. Ghatak, S. Dutta, B. Chatterjee, and M. Gupta, “A survey on comparison analysis between EEG signal and MRI for brain stroke detection,” in *Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2018, Volume 3*, 2019, pp. 377–382.

[26] Y.-A. Choi *et al.*, “Machine-learning-based elderly stroke monitoring system using electroencephalography vital signals,” *Appl. Sci.*, vol. 11, no. 4, p. 1761, 2021.

[27] M. Kaur, S. R. Sakhare, K. Wanjale, and F. Akter, “Early stroke prediction methods for prevention of strokes,” *Behav. Neurol.*, vol. 2022, 2022.

[28] A. Sawan, M. Awad, and R. Qasrawi, “Machine learning-based approach for stroke classification using electroencephalogram (EEG) signals,” 2022.

[29] Wikipedia, “Engine Control Unit,” *Wikipedia*, pp. 1–7, 2021, [Online]. Available: https://en.wikipedia.org/w/index.php?title=Special:DownloadAsPdf&page=Engine_control_unit&action=show-download-screen

[30] E. Dritsas and M. Trigka, “Stroke risk prediction with machine learning techniques,” *Sensors*, vol. 22, no. 13, p. 4670, 2022.

[31] E. R. Kandel, J. H. Schwartz, T. M. Jessell, S. Siegelbaum, A. J. Hudspeth, and S. Mack, *Principles of neural science*, vol. 4. McGraw-hill New York, 2000.

[32] “Electroencephalography (2).pdf.crdownload.”

[33] “EEG - Electroencephalogram | KidsHealth NZ.” https://www.kidshealth.org.nz/eeg-electroencephalogram (accessed May 26, 2023).

[34] T. Mulholland, “Objective EEG methods for studying covert shifts of visual attention,” in *The psychophysiology of thinking*, Elsevier, 1973, pp. 109–151.

[35] T. Burns and R. Rajan, “Combining complexity measures of EEG data: multiplying measures reveal previously hidden information,”

F1000Research, vol. 4, no. 137, p. 137, 2015.

[36] J. N. Acharya and V. J. Acharya, “Overview of EEG montages and principles of localization,” *J. Clin. Neurophysiol.*, vol. 36, no. 5, pp. 325–329, 2019.

[37] R. Srinivasan, “Methods to improve the spatial resolution of EEG,” *Int. J. Bioelectromagn.*, vol. 1, no. 1, pp. 102–111, 1999.

[38] P. L. Nunez and K. L. Pilgreen, “The spline-Laplacian in clinical neurophysiology: a method to improve EEG spatial resolution,” *J. Clin. Neurophysiol.*, vol. 8, no. 4, pp. 397–413, 1991.

[39] M. Ziyad, “Artificial Intelligence Definition, Ethics and Standards,” *Artif. Intell. Defin. Ethics Stand.*, pp. 1–11, 2019.

[40] M. V Valueva, N. N. Nagornov, P. A. Lyakhov, G. V Valuev, and N. I. Chervyakov, “Application of the residue number system to reduce hardware costs of the convolutional neural network implementation,” *Math. Comput. Simul.*, vol. 177, pp. 232–243, 2020.

[41] I. El Naqa and M. J. Murphy, “What Is Machine Learning?,” *Mach. Learn. Radiat. Oncol.*, pp. 3–11, May 2015, doi: 10.1007/978-3-319-18305-3_1.

[42] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2018.

[43] A. Ratner, S. Bach, P. Varma, and C. Ré, “Weak supervision: the new programming paradigm for machine learning,” *Hazy Res. Available via https//dawn. cs. stanford. edu//2017/07/16/weak-supervision/. Accessed*, vol. 5, 2019.

[44] “Machine learning”.

[45] “What Is the Definition of Machine Learning? | Expert.ai | expert.ai.”

https://www.expert.ai/blog/machine-learning-definition/ (accessed Jun. 22, 2023).

[46] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations,” in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 609–616.

[47] V. Cortes, “Cortes c., vapnik v,” *Support. networks, Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[48] R. Metz, “Why Microsoft’s teen chatbot, Tay, said lots of awful things online.” MIT Technology Review. Retrieved from https://www. technologyreview. com …, 2016.

[49] “ML | Understanding Data Processing - GeeksforGeeks.” https://www.geeksforgeeks.org/ml-understanding-data-processing/ (accessed Jun. 22, 2023).

[50] I. Young, J. Gerbrands, and L. van Vliet, “Fundamentals of Image Processing,” pp. 1–85, 2009, doi: 10.1201/9781420046090-c13.

[51] “ML | Introduction to Data in Machine Learning - GeeksforGeeks.” https://www.geeksforgeeks.org/ml-introduction-data-machine-learning/ (accessed Jun. 22, 2023).

[52] B. Jähne, “Digital Image Processing 5th revised and extended edition,” *Berlin Springer-Verlag, doi*, vol. 10, pp. 3–540, 2002.

[53] B. Jähne, *Digital Image Processing, 5th revised and extended edition*, vol. 13, no. 9. 2002. doi: 10.1088/0957-0233/13/9/711.

[54] S. Vijayaraghavan, “LECTURE NOTES BE (IVYEAR)”.

[55] D. Bradley and G. Roth, “Adaptive Thresholding using the Integral

Image," *J. Graph. Tools*, vol. 12, no. 2, pp. 13–21, 2007, doi: 10.1080/2151237x.2007.10129236.

[56] "ImageJ: Thresholding — Introduction to Bioimage Analysis." https://bioimagebook.github.io/chapters/2-processing/3-thresholding/imagej.html (accessed Jul. 15, 2023).

[57] S. Zachow, M. Zilske, and H.-C. Hege, "3D reconstruction of individual anatomy from medical image data: Segmentation and geometry processing," 2007.

[58] D. Guo, Y. Pei, K. Zheng, H. Yu, Y. Lu, and S. Wang, "Degraded image semantic segmentation with dense-gram networks," *IEEE Trans. Image Process.*, vol. 29, pp. 782–795, 2019.

[59] J. Yi, P. Wu, M. Jiang, Q. Huang, D. J. Hoeppner, and D. N. Metaxas, "Attentive neural cell instance segmentation," *Med. Image Anal.*, vol. 55, pp. 228–240, 2019.

[60] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9404–9413.

[61] C. Mouton, J. C. Myburgh, and M. H. Davel, "Stride and translation invariance in CNNs," in *Artificial Intelligence Research: First Southern African Conference for AI Research, SACAIR 2020, Muldersdrift, South Africa, February 22-26, 2021, Proceedings 1*, 2020, pp. 267–281.

[62] C. Strobl, A. L. Boulesteix, and T. Augustin, "Unbiased split selection for classification trees based on the Gini index," *Comput. Stat. Data Anal.*, vol. 52, no. 1, pp. 483–501, Sep. 2007, doi: 10.1016/j.csda.2006.12.030.

[63] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional

architecture of monkey striate cortex," *J. Physiol.*, vol. 195, no. 1, pp. 215–243, 1968.

[64] "Architecture of LeNet-5 (LeCun et al., 1998). | Download Scientific Diagram." https://www.researchgate.net/figure/Architecture-of-LeNet-5-LeCun-et-al-1998_fig2_317496930 (accessed Jul. 21, 2023).

[65] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998, doi: 10.1109/5.726791.

[66] M. M. Taye, "Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions," *Computation*, vol. 11, no. 3, 2023, doi: 10.3390/computation11030052.

[67] A. Ghosh, A. Sufian, F. Sultana, A. Chakrabarti, and D. De, "Fundamental concepts of convolutional neural network," *Recent trends Adv. Artif. Intell. Internet Things*, pp. 519–567, 2020.

[68] A. Gramfort *et al.*, "MEG and EEG data analysis with MNE-Python," *Front. Neurosci.*, no. 7 DEC, 2013, doi: 10.3389/fnins.2013.00267.

[69] "OpenNeuro." https://openneuro.org/ (accessed Jul. 03, 2023).

[70] "PhysioNet." https://physionet.org/ (accessed May 27, 2023).

[71] J. W. Kantelhardt, S. A. Zschiegner, E. Koscielny-Bunde, S. Havlin, A. Bunde, and H. E. Stanley, "Multifractal detrended fluctuation analysis of nonstationary time series," *Phys. A Stat. Mech. its Appl.*, vol. 316, no. 1–4, pp. 87–114, 2002, doi: 10.1016/S0378-4371(02)01383-3.

[72] S. Finnigan, A. Wong, and S. Read, "Defining abnormal slow EEG activity in acute ischaemic stroke: Delta/alpha ratio as an optimal QEEG index," *Clin. Neurophysiol.*, vol. 127, no. 2, pp. 1452–1459, 2016.

[73] Z. Liu, J. Sun, Y. Zhang, and P. Rolfe, “Sleep staging from the EEG signal using multi-domain feature extraction,” *Biomed. Signal Process. Control*, vol. 30, pp. 86–97, 2016.

[74] W. R. W. Omar, Z. Mohamad, M. N. Taib, and R. Jailani, “ANN classification of ischemic stroke severity using EEG sub band relative power ration,” in *2014 IEEE Conference on Systems, Process and Control (ICSPC 2014)*, 2014, pp. 157–161.

[75] V. Matic *et al.*, “Objective differentiation of neonatal EEG background grades using detrended fluctuation analysis,” *Front. Hum. Neurosci.*, vol. 9, p. 189, 2015.

[76] P. Singh, S. D. Joshi, R. K. Patney, and K. Saha, “Fourier-based feature extraction for classification of EEG signals using EEG rhythms,” *Circuits, Syst. Signal Process.*, vol. 35, pp. 3700–3715, 2016.

[77] J. Brownlee, “Tactics to combat imbalanced classes in your machine learning dataset,” *Mach. Learn. Mastery*, vol. 19, 2015.

[78] D. S. Blogathon, N. Fraud, and N. Fraud, “5 Techniques to Handle Imbalanced Data For a Classification Problem What is imbalanced data ? Approach to deal with the imbalanced dataset problem”.

[79] H. M. Models, “Time-series segmentation”.

[80] X. Jiang and G. Bian, “Removal of Artifacts from EEG Signals : A Review,” pp. 1–18, 2019, doi: 10.3390/s19050987.

[81] “What? When? How?: ExtraTrees Classifier | by Karun Thankachan | Towards Data Science.” https://towardsdatascience.com/what-when-how-extratrees-classifier-c939f905851c (accessed May 27, 2023).

[82] S.-A. Ouadfeul and L. Aliouane, “Hurst Exponent,” pp. 1–6, 2022, doi:

10.1007/978-3-030-26050-7_154-1.

[83] “Hurst exponent in Python using the Hurst module - CodeSpeedy.” https://www.codespeedy.com/hurst-exponent-in-python/ (accessed Jul. 23, 2023).

[84] S. Finnigan, A. Wong, and S. Read, “Defining abnormal slow EEG activity in acute ischaemic stroke: Delta/alpha ratio as an optimal QEEG index,” *Clin. Neurophysiol.*, vol. 127, no. 2, pp. 1452–1459, 2016, doi: 10.1016/j.clinph.2015.07.014.

[85] P. Welch, “The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms,” *IEEE Trans. audio Electroacoust.*, vol. 15, no. 2, pp. 70–73, 1967.

[86] “Multilayer perceptron - Wikipedia.” https://en.wikipedia.org/wiki/Multilayer_perceptron (accessed May 27, 2023).

[87] “Random Forest Algorithms - Comprehensive Guide With Examples.” https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/ (accessed Jul. 22, 2023).

[88] D. Amos, D. (Computer software developer) Bader, J. Jablonski, and F. Heisler, *Python basics : a practical Introduction to Python 3 FOURTH EDITION*. 2021.

[89] “بابل مستشفى الحله التعليمي - Al Hillah - WorldPlaces.” https://iraq.worldplaces.me/places-in-al-hillah/24219217.html (accessed Jul. 03, 2023).

[90] “مستشفى الإمام الصادق التعليمي (المستشفى التركي) Map - Hospital - Iraq - Mapcarta.”

https://mapcarta.com/N1821569533 (accessed Jul. 22, 2023).

[91] S. E. E. Profile, “Detection of Brain Stroke from CT Scan Image,” no. June 2018, 2018, doi: 10.13140/RG.2.2.26552.49922.

[92] B. G. Batchelor and F. M. Waltz, “Morphological image processing,” *Mach. Vis. Handb.*, no. November, pp. 802–870, 2012, doi: 10.1007/978-1-84996-169-1_19.

[93] N. Otsu and D. Analysis, “Otsu ’ s method”.

[94] “Morphological Image Processing.” https://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/topic4.htm (accessed May 31, 2023).

[95] R. Al-Alwani, P.-D. K. Kasim Abdalla, F. Nabil Abbas, and A. Affiliations, “Brain Stroke Detection Using Image Processing,” vol. 38, no. 01, pp. 703–711, 2023.

[96] “Mathematical morphology - Wikipedia.” https://en.wikipedia.org/wiki/Mathematical_morphology (accessed May 31, 2023).

[97] “Otsu’s method - Wikipedia.” https://en.wikipedia.org/wiki/Otsu%27s_method (accessed May 31, 2023).

[98] A. Qureshi, R. Zheng, A. Ahmed Qureshi, C. Zhang, and A. Elmeligi, “Ischemic Stroke Detection using EEG Signals,” *Cascon*, no. October 2018, pp. 301–308, 2018, [Online]. Available: https://www.researchgate.net/publication/330599733

[99] easyai.tech, “Random Forest – Random forest,” *2018 Second Int. Conf. Intell. Comput. Control Syst.*, no. Iciccs, pp. 587–588, 2019, [Online]. Available: https://easyai.tech/en/ai-definition/random-forest/

[100] R. Kumar and A. Indrayan, “Receiver operating characteristic (ROC) curve for medical researchers,” *Indian Pediatr.*, vol. 48, no. 4, pp. 277–287, 2011, doi: 10.1007/s13312-011-0055-4.

[101] K. M. Ting, “Confusion Matrix,” *Encycl. Mach. Learn. Data Min.*, pp. 260–260, 2017, doi: 10.1007/978-1-4899-7687-1_50.

[102] E. R. A. Al-alwani, “Brain stroke detection using ANN based on EEG signals,” vol. 2017.

[103] R. Al-alwani, K. K. Abdalla, and F. N. Abbas, “Brain Stroke Detection using MRI Image and Logistic Regression,” vol. 38, no. 01, pp. 703–711, 2023.

[104] V. Krishna, J. S. Kiran, P. P. Rao, G. C. Babu, and G. J. Babu, “Early detection of brain stroke using machine learning techniques,” in *2021 2nd International Conference on Smart Electronics and Communication (ICOSEC)*, 2021, pp. 1489–1495.

Appendix (A)

Feature Extraction Program --- Source Code -- Python List

Appendix-A

The source code program/ feature 99.py)

```
# Feature 99.py // extract the attributes from the EEG signals and create the #file csv represent all features in time and frequency domain

import csv
import math
import sns as sns
import yasa
import mne_features
from scipy import signal
import pandas
import mne
import matplotlib.pyplot as plt
import json
import pandas as pd
import numpy as np
import datetime
from datetime import datetime, timedelta
from hurst import compute_Hc
from scipy.signal import welch
import seaborn as sns
import pandas as pd
import os

# read all data from the files set,tsv,json
#load all data file to the particpient -1
# load set files only from ds2691 that contain 32 channel there are 20 files represant 20 participant
# \\\\ we collect 4 ch only ch30,ch29,ch26,ch25 \\\\\
#patient=('F:\EEG DATA\EEG DATA -OUT SIDE/1-ds002691-set-ok\O-DS002691\sub-001\egg1\sub001.set')
#patient=('F:\EEG DATA\EEG DATA -OUT SIDE/1-ds002691-set-ok\O-DS002691\sub-002\eeg/sub002.set')
#patient=('F:\EEG DATA\EEG DATA -OUT SIDE/1-ds002691-set-ok\O-DS002691\sub-003\eeg\sub003.set')
#patient=('F:\EEG DATA\EEG DATA -OUT SIDE/1-ds002691-set-ok\O-DS002691\sub-004\eeg\sub004.set')

#patient=('F:\EEG DATA\EEG DATA -OUT SIDE/1-ds002691-set-ok\O-
DS002691\sub-005\eeg\sub005.set')
#patient=('F:\EEG DATA\EEG DATA -OUT SIDE/1-ds002691-set-
ok\O-DS002691\sub-006\eeg\sub006.set'
#patient=('F:\EEG DATA\EEG DATA -OUT SIDE/1-ds002691-set-ok\O-
DS002691\sub-007\eeg\sub007.set')
#patient=('F:\EEG DATA\EEG DATA -OUT SIDE/1-ds002691-set-
patient=('F:\EEG DATA\EEG DATA -OUT SIDE/1-ds002691-set-ok\O-
DS002691\sub-009\eeg\sub009.set')
#patient=('F:\EEG DATA\EEG DATA -OUT SIDE/1-ds002691-set-ok\O-
DS002691\sub-010\eeg\sub010.set')
# patient=('F:\EEG DATA\EEG DATA -OUT SIDE/1-ds002691-set-ok\R-
DS0002691\sub-011\eeg\yy11.set')
#patient=('F:\EEG DATA\EEG DATA -OUT SIDE/1-ds002691-set-ok\R-
DS0002691\sub-012\eeg\yy12.set')
#patient=('F:\EEG DATA\EEG DATA -OUT SIDE/1-ds002691-set-ok\O-
DS002691\sub-013\eeg\sub013.set')
# patient=('F:\EEG DATA\EEG DATA -OUT SIDE/1-ds002691-set-ok\O-
DS002691\sub-014\eeg\sub014.set')
# patient=('F:\EEG DATA\EEG DATA -OUT SIDE/1-ds002691-set-ok\O-
DS002691\sub-015\eeg\sub015.set')
# patient=('F:\EEG DATA\EEG DATA -OUT SIDE/1-ds002691-set-ok\O-
DS002691\sub-016\eeg\sub016.set')
# patient=('F:\EEG DATA\EEG DATA -OUT SIDE/1-ds002691-set-ok\O-
DS002691\sub-017\eeg\sub017.set')
# patient=('F:\EEG DATA\EEG DATA -OUT SIDE/1-ds002691-set-ok\O-
DS002691\sub-018\eeg\sub018.set')
# patient=('F:\EEG DATA\EEG DATA -OUT SIDE/1-ds002691-set-ok\O-
DS002691\sub-019\eeg\sub019.set')
# patient=('F:\EEG DATA\EEG DATA -OUT SIDE/1-ds002691-set-ok\O-
DS002691\sub-020\eeg\sub020.set')

#read json files ------------------------------------
# gg2=open('F:\EEG DATA\EEG DATA -OUT SIDE/1-ds002691-set-ok\O-
DS002691\sub-001\egg1\eeg-js.json','r')
# gg3=open('F:\EEG DATA\EEG DATA -OUT SIDE/1-ds002691-set-ok\O-
DS002691\dataset_description.json','r')
# gg4=open('F:\EEG DATA\EEG DATA -OUT SIDE/1-ds002691-set-ok\O-
DS002691\participants.json','r')
# gg5=open('F:\EEG DATA\EEG DATA -OUT SIDE/1-ds002691-set-ok\O-
DS002691/task_motorimagery_eeg.json','r')
```

```
# gg6=open('F:\EEG DATA\EEG DATA -OUT SIDE/1-ds002691-set-ok\O-
DS002691/task_motorimagery_nirs.json','r')
# # -------- read the tsv files =================
# hh1=pd.read_csv('F:\EEG DATA\EEG DATA -OUT SIDE/1-ds002691-
set-ok\O-DS002691\sub-001\egg1\cha_ts.tsv',
#sep ='\t')
# hh2=pd.read_csv('F:\EEG DATA\EEG DATA -OUT SIDE/1-ds002691-
set-ok\O-DS002691\participants.tsv',
#sep ='\t')
# hh3=pd.read_csv('F:\EEG DATA\EEG DATA -OUT SIDE/1-ds002691-
set-ok\O-DS002691\sub-001\egg1\sub001-event-ts.tsv',
#sep ='\t')
# print('the tsv file ',hh1)
nn1=patient
print('the path ', patient)
nn2=nn1[-10:]
print('file name ',nn2)
# ////////////////////////////////
raw=mne.io.read_raw_eeglab(patient) # the main file set extansion // source
file ///////////////////////
raw.filter(0.5,45)# filter to reject all freq gtater then 45 hz
print('freq filter',raw.filter)
print('the raw \n: ',raw)
print('the raw info\n : ', raw.info)
fs_samp=raw.info['sfreq']
print('samplig freq',fs_samp)
#print('the raw head',raw.)
raw2=raw.crop(tmax=60) # ///// comp the time // first step cmpreesion the
data //////
print('the raw after comp\n',raw2)
print('the raw len ', len(raw2))
raw_tamp=raw2.copy()
raw_tamp.drop_channels(['E1','E2','E3','E4','E5','E6','E7','E8','E9','E10','
E11','E12','E13','E14','E15','E16','E17'
,'E18','E19','E20','E21','E22','E23','E24','E27','E28','E31','E32'])
print('the raw temp',raw_tamp.info)
CHMM=raw_tamp.ch_names
print('the names',CHMM)
data=raw2.get_data()
#raw_x=raw2.copy()
#zzr=raw_x.drop_channels(['E1','E2','E3','E4','E5','E6','E7','E8','E9',''])
#dat2=zzr.get_data()
```

```
print('the raw after drop',data.shape)
#datadrop_channels(['E1','E2'])
#print('the vv shape',vv.shape)
# E30X=raw2['E30']

# print('e30',([E30X][0][1]))

#BB5=raw2.get_data(picks=0)
print()
#print('the data',data.shape)# data shape 32x15001
#print('the e1',len(raw2['E4']))
ALL_4CH=raw2.get_data(picks=['E25','E26','E29','E30']) #%get 4 channels only
#print('all_4ch',ALL_4CH.shape) #// get 4x15001 data shape
#print(ALL_4CH.info)
ch29=raw2.get_data(picks=['E29']) # piks channel E29 --- FRONT
print('ch29\n',ch29.shape)
print('the len',len(ch29))
ch30=raw2.get_data(picks=['E30']) # pikes ch E30---- FRONT
print('ch30\n',ch30.shape)
ch25=raw2.get_data(picks=['E25']) # PICKS CH E 25 ---- BACK
print('ch25 \n',ch25.shape)
ch26=raw2.get_data(picks=['E26']) # PICK CH 26 --- BACK
print('ch26',ch26.shape)
print()
print()
#get tHE TRANSPOSE FOR EACH CHANEEL
ch30T=ch30.transpose()
ch29T=ch29.transpose()
ch25T=ch25.transpose()
ch26T=ch26.transpose()
print('transpose',ch30T.shape)
print('ch30',len(ch30T))
# ////////// calculate the hurst with mean for each chanel ,USE IST FOR
ch29H=mne_features.univariate.compute_hurst_exp(ch29) # the hurst value
ch29m=mne_features.univariate.compute_mean(ch29)# the mean value
ch30H=mne_features.univariate.compute_hurst_exp(ch30)
ch30m=mne_features.univariate.compute_mean(ch30)
ch25H=mne_features.univariate.compute_hurst_exp(ch25)
ch25m=mne_features.univariate.compute_mean(ch25)
ch26H=mne_features.univariate.compute_hurst_exp(ch26)
ch26m=mne_features.univariate.compute_mean(ch26)
```

```
#// PRINT OUT THE RESULT USING 1 FOR
print('hurst+ mean to 4 cha using 1 st form ')
print('ch29 mean ',ch29m) # the hurst
print('ch29 hurst ',ch29H) # the mean
print('ch30 mean ',ch30m)
print('ch30 hurst ',ch30H)
print('ch25mean ',ch25m)
print('ch25 hurst ',ch25H)
print('ch26 mean ',ch26m)
print('ch26 hurst ',ch26H)
print()
print()

# CALCULETE THE HURST USING THE FUNCTION compute_HC()
AND PRINT OUT///////////////////
print(nn2,'/// hurat using 2nd form /// ')
harr=[] # ////////////ADD THE harr array

H30,c30,val30=compute_Hc(ch30T)
H30=format(H30,'.4f')
print ('the hurs-ch30=',H30)

H29,c29,val29=compute_Hc(ch29T)
H29=format(H29,'.4f')
print ('the hurs-ch29=',H29)

H26,c26,val26=compute_Hc(ch26T)
H26=format(H26,'.4f')
print ('the hurs-ch26=',H26)

H25,c25,val25=compute_Hc(ch25T)
H25=format(H25,'.4f')
print ('the hurs-ch25=',H25)

harr.append(H25)
harr.append(H26)
harr.append(H29)
harr.append(H30)
print()
print()
```

APPANDIX -A --Feature Extraction Program --- Source Code -- Python List

```
## //// Extracting data by index //////from a12.py ////////////////////

print('Extracting data by index & plot' )
sampling_freq = raw2.info['sfreq']
print('the feq',sampling_freq)
start_stop_seconds = np.array([11,13]) # max no index =60, equal=
60x250=15000sample
print('start stop ',start_stop_seconds) # gg1= 600.61x 52,,,
start_sample, stop_sample = (start_stop_seconds *
sampling_freq).astype(int)
print('start',start_sample)
print('stop ',stop_sample)
col=[30,29,26,25]
for c in col:
channel_index= c # no of channel max =376
raw_selection= raw[channel_index, start_sample:stop_sample]
#print('len raw.selection',len(raw_selection)) # raw-selaction [0],[1]


xd1=raw_selection[1]
yd1=raw_selection[0].T
if c==30:
 plt.plot(xd1,yd1)
 plt.title(nn2,x=0.2 ,y=1)
 plt.title ('fig-1-ch-30', loc='right')
 plt.show()
elif c==29:
plt.plot(xd1, yd1)
plt.title (nn2,x=0.2,y=1)
plt.title('fig-2-ch- E29',loc='right')
plt.show()
elif c== 26:
plt.plot (xd1, yd1)
plt.title (nn2, x=0.2, y=1)
plt.title('fig-3-ch- E26',loc='right')
plt.show()
elif c== 25:
plt.plot (xd1, yd1)
plt.title (nn2, x=0.2, y=1)
plt.title ('fig-4-ch- E25',loc='right')
plt.show ()
```

```
else:

print('out out')


# /// Extracting channels by name //////// from a12.py //////////
print('Extracting channels by name ')
channel_names = ['E30', 'E29']
print('Channel names ',channel_names)
two_eeg_chans = raw[channel_names, 3750:7000]
y_offset = np.array([5e-11, 0]) # just enough to separate the channel traces
x = two_eeg_chans[1]
y = two_eeg_chans[0].T+16
#plt.figure(figsize=(9,12))
lines = plt.plot(x, y)
plt.legend(lines, channel_names)
plt.title(nn2,x=0.2, y=1)
plt.title('figure-5',loc='right') # fig 5 //////////////////////
plt.show()

# ///////compute the mean ,std, median,
print('store the values 1x4 mean val to 4 ch ')
hkk=[]
ch29_mean=np.mean(ch29)
ch29_median =np.median(ch29)
ch29_std=np.std(ch29)
ch30_mean=np.mean(ch30)
ch30_median =np.median(ch30)
ch30_std=np.std(ch30)
ch26_mean=np.mean(ch26)
ch26_median =np.median(ch26)
ch26_std=np.std(ch26)
ch25_mean=np.mean(ch25)
ch25_median =np.median(ch25)
ch25_std=np.std(ch25)
hkk.append(ch25_mean)
hkk.append(ch26_mean)
hkk.append(ch29_mean)
hkk.append(ch30_mean)
print()
```

```
# ///// print the result mean, std, median ///////////
staa=[]
print(nn2,'//the statistical cal 0f all 4 signal diractly//')

print('ch 29 mean',ch29_mean)

print('ch 29 median',ch29_median)
print('ch 29 std',ch29_std)
print('ch 30 mean',ch30_mean)
print('ch 30 median',ch30_median)
print('ch 30 std',ch30_std)
print('ch 26 mean',ch26_mean)
print('ch 26 median',ch26_median)
print('ch 26 std',ch26_std)
print('ch 25 mean',ch25_mean)
print('ch 25 median ',ch25_median)
print('ch 25 std',ch25_std)
staa.append(ch30_mean)
staa.append(ch29_mean)
staa.append(ch26_mean)
staa.append(ch25_mean)
print()
print()

# ///////////calculate the welch power //////////////////

print(nn2,'//calcu the welch power //')

#/////////////////////////////////////////////////////
# m1=[]
# m2=[7,7,9,52]
# ch=['pp1','pp9','99',8]
# str1=[]
# for i,j,k in zip(ch,m1,m2):
#print(i,j,k)
#print('ooopp')
# print('op000')
# print()
x1=[]
x2=[]
ch=[ch25,ch26,ch29,ch30]
m1=['p-ch25m','p-ch26m','p-ch29m','p-ch30m']
m2=['p-ch25mid','p-ch26mid','p-ch29mid','p-ch30mid']
```

APPANDIX -A --Feature Extraction Program --- Source Code -- Python List

The source code program

```
str1=[0,1,2,3]

str2=[0,1,2,3]
print(' start start ')

for i,j,k,z1,z2 in zip(ch,m1,m2,str1,str2): # creat array to save the mean
power x1, creat the median array x2
#print(' the j,k',j,k)
x,y=signal.welch(i,fs_samp,nperseg=1024) #// ch 29- //////////////////
x1.append(np.mean(y[:512])) # save the mean values in x1
x2.append(np.median(y[:512])) #save the median power in x2
## #print('the f-ch29w\n',len(f_ch29w),'\n p-ch29w\n',len(p_ch29w))
print(j,x1[z1],'\n',k,x2[z2])
# print('//// out loop /// ')
## print('ppp44')
print()
print()
print('\\\\\\ppppp\\\\\\\\\\\\\\')

print('show the mean\wwlch and median\welch valce')

# print()
# for i in str1:
#print('iuuu',x1[i])
for i,j in zip(str1,str2):
 print('the x1 mean val:',x1[i])
 print('the x2 median ',x2[j])
 print('loop')
print()
print()


# #////////////////////////////////////////////


# //// plot welch for all 4 signales ///////////
# // plot ch29 ////\\\\\\\\\\\\\\\\\\\\\\

k= [ch30,ch29,ch26,ch25]# ch array\\\\ plot
m=['fig-6-ch30','fig-7-ch29','fig-8-ch26','fig-9-ch25'] # string array
```

```
for i,j in zip(k,m):
 fg,pg= signal.welch (i, fs_samp, nperseg=1024)
 pg=pg.transpose()
 plt.semilogy(fg,pg)
 plt.xlim([0,100])
 plt.xlabel('freq in hz ')
 plt.ylabel('power')

 plt.title(nn2,x=0.2,y=1)
 plt.title(j,loc='right')
 plt.show()
print()


#\\\\\\\\\\\\\\\\\\\\\ calculate the band power for delta,theta, alpha ////\\\\\\\\
# //// TO FOR CH 29, 30,26,25 ///////////
 # FIRST APPLAY A BANDBASS FILER

print(' //// calculation power band frq for each channel ///')
# there are two waye 1-using yasa function 2- using each indiv cha
# to calculate -the delta band, theta freq band, beta freq for each channels
,so we get 12 features
# each ch 3 feture ,delat,theta, beta
# 1- method a --- using yasa
# first calculat the welch, cal the feq band


sf=250
#print('all-4ch',ALL_4CH.shape)
datau=raw_tamp.get_data()
print('the datau',datau.shape)
win = int(4 * sf) # Window size is set to 4 seconds
fr1, psd1 = welch(datau, sf, nperseg=win, average='median')
print('the fr1\n',fr1.shape,'\n the psd1\n',psd1.shape,'\n')

# plot the
# //// plot //// plot ////
#chan=raw2.ch_names
#print('the chan',ALL_4CH.shape)
```

```
plt.plot(fr1, psd1[1], 'k', lw=2)
plt.fill_between(fr1, psd1[3], cmap='Spectral')
plt.xlim(0, 50)
plt.yscale('log')
sns.despine()
plt.suptitle(nn2,x=0.2,y=1)
plt.title(CHMM[3],loc='right')
plt.title('fig-10')
plt.xlabel('Frequency [Hz]')
plt.ylabel('PSD log($uV^2$/Hz)')
plt.show()

# Relative power
#
# Note that the TotalAbsPow column contains the total absolute (physical)
power, summed across all bands.

# Relative power: sum of all (non-overlapping and sequential) bands equals
to 1
tabel=yasa.bandpower_from_psd(psd1, fr1, ch_names=CHMM)
# cal relative band power

print('the tabel cal the eeg bandpowe \n',tabel) # print all values using yasa
# cal relative band power
print('\\\\cal relative band power\\\\')
#print('THE DELTA',tabel['Delta'])
yx=[] # store all vau in y1 array (power pands )
zx0=[0,1,2,3,4,5,6,7,8,9,10,11]
xc1=[tabel['Delta'][0],tabel['Theta'][0],tabel['Alpha'][0],tabel['Delta'][1],tab
el['Theta'][1],tabel['Alpha'][1],
tabel['Delta'][2],tabel['Theta'][2],tabel['Alpha'][2],tabel['Delta'][3],tabel['T
heta'][3],tabel['Alpha'][3]]

xc2=[tabel['Theta'][0],tabel['Delta'][0],tabel['Delta'][0],tabel['Theta'][1],tab
el['Delta'][1],tabel['Theta'][1],
tabel['Theta'][2],tabel['Delta'][2],tabel['Delta'][2],tabel['Theta'][3],tabel['D
elta'][3],tabel['Delta'][3]]

xc3=[tabel['Alpha'][0],tabel['Alpha'][0],tabel['Theta'][0],table
['Alpha'][1],tabel['Alpha'][1],tabel['Delta'][1],
tabel['Alpha'][2],tabel['Alpha'][2],tabel['Theta'][2],tabel['Alpha'][3],tabel['
Alpha'][3],tabel['Theta'][3]]
zx4=['ch25-p-delta','ch25-p-theta','ch25-p-alpha','ch26-p-delta','ch26-p-
theta','ch26-p-alpha','ch29-p-delta','ch29-p-theta',
'ch29-p-alpha','ch30-p-delta','ch30-p-theta','ch30-p-alpha']
for i,j,k,m,l in zip(xc1,xc2,xc3,zx0,zx4):
 a=i/(i+j+k)
 yx.append(a)
 print(l,yx[m],'\n')
 print('pass')
print('out out out ')
print()
# cal the relative hemisp
print('cal relative hemispare first mathod ')
y2=[]
zx1=[tabel['Delta'][0],tabel['Theta'][0],tabel['Alpha'][0],tabel['Theta'][2],ta
bel['Alpha'][2],tabel['Alpha'][2]]
zx2=[tabel['Delta'][1],tabel['Theta'][1],tabel['Alpha'][1],tabel['Theta'][3],ta
bel['Alpha'][3],tabel['Alpha'][3]]
zx3=[0,1,2,3,4,5]
for i,j,k in zip(zx1,zx2,zx3):
 a=(i-j)/(i+j)
 y2.append(a)
 print(y2[k])
 print('passss')
print('out 44')
print() #space that set in the o/p
print()
print('////using a new way ///')
#//using new way to calculate the power band
# calculate -relitace power band ratio for each delta,theta, alpha, for each
chanel 12 item
# cal the relitave pwer ratio / hemispher
# first cal the power using welch for each ch
print('/// cal the powe band using mehod 2 //// ')
win=4*sf
fr25,psd25=signal.welch(ch25,sf,nperseg=win)
psd25t=psd25.transpose()
#print('the fr1\n',fr25.shape,'the psd25\n',psd25.shape,'\n')
```

```
fr26,psd26=signal.welch(ch26,sf,nperseg=win)
psd26t=psd26.transpose()
fr29,psd29=signal.welch(ch29,sf,nperseg=win)
psd29t=psd29.transpose()
fr30,psd30=signal.welch(ch30,sf,nperseg=win)
psd30t=psd30.transpose()
# print('the oo',psd30.shape)
# print('the oiu',psd30t.shape)
ch25_f= psd25.flatten()
ch26_f= psd26.flatten()
ch29_f= psd29.flatten()
ch30_f= psd30.flatten()

fr25t=fr25.transpose()
fr26t=fr26.transpose()
fr29t=fr29.transpose()
fr30t=fr30.transpose()
########################################3#######
# # //// plot each channels ch25,ch26,ch29,ch30/////////////////////////
 #Define window length (4 seconds)
pk1=[fr25,fr26,fr29,fr30]
pk2=[psd25t,psd26t,psd29t,psd30t]
pk77=[ch25_f,ch26_f,ch29_f,ch30_f]
pk88=[psd25,psd26,psd29,psd30]
pk3=['fig-11-welch-plot-ch25','fig-12-welch-plot-26','fig-13-welch-plot-
ch29','fig-14-welch-plot-ch30']
#pk5=['welch-plot-ch25-pwr','welch-plot-26-pwr','welch-plot-
ch29pwr','welch-plot-ch30-pwr']
#pk4=[0.5,4,8]
frl=[0.5,4,8]
frh=[4,8,12]
for i,j,k in zip(pk1,pk2,pk3): # plot all channels welch power fig-11 --- fig -
14
 sns.set(font_scale=1.2, style='white') # plot from fig 11 into fig 14
 plt.figure(figsize=(8, 5))
 plt.plot(i,j, color='k', lw=2)
 plt.xlabel('Frequency (Hz)')
 plt.ylabel('Power spectral density (V^2 / Hz)')
 plt.ylim([0, j.max() * 1.1])
 plt.title(nn2,x=0.25, y=1)
 plt.title(k,loc='right')
 plt.xlim([0,i.max()])
sns.despine()
 plt.show()
print('exit exit from plot ')
print()
#################################################################

# cal the powerbands to each channels ch25,ch26,ch29,30 with bands delta,theta, alpha and plot
for p1,p2 in zip(frl,frh):
low,high=p1,p2

if low==0.5:
pk5=['fig-15-ch25-delta','fig-16-ch26-delta','fig-17-ch29-delta','fig-18-ch30-delta']
if low==4:
pk5= ['fig-19-ch25-theta', 'fig-20-ch26-theta', 'fig-21-ch29-theta','fig-22-ch30-theta']

if low==8:
pk5= ['fig-23-ch25-alpha', 'fig-24-ch26-alpha', 'fig-25-ch29-alpha','fig-26-ch30-alpha']

for i,j,k,z in zip(pk1,pk2,pk77,pk5):

 idx_d = np.logical_and (i>= low,i<= high)
# Plot the power spectral density and fill the delta area/
 plt.figure (figsize=(7, 5))
 plt.plot (i, j, lw=2, color='k')
#res = psd25.flatten()
 plt.fill_between (i,k, where=idx_d, color='skyblue')
 plt.xlabel ('Frequency (Hz)')
 plt.ylabel ('Power spectral density (uV^2 / Hz)')
 plt.xlim ([0, 14])
 plt.ylim ([0, j.max () * 1.1])
 plt.title(nn2,x=0.2,y=1)
 plt.title (z,loc='right')
 sns.despine()
 plt.show()
```

```
#########################################################
# cal the bandpoer using 2nd method (relitive
arr1=[] # delta band to 4 ch 1x4
arr2=[]
arr3=[]
pxu=[0,1,2,3]
print(nn2,'//print out the values of power band new way//')
from scipy.integrate import simps
#pxt=[arr1,arr2,arr3]
for p1,p2 in zip(frl,frh):
low,high=p1,p2
if low==0.5:
#px6=[arr1,arr1,arr1,arr1]
pxt=['ch25-delta-abp','ch26-delta-abp','ch29-delta-abp','ch30-delta-abp']
if low==4:
#px6=arr2
pxt = ['ch25-theta-abp', 'ch26-theta-abp', 'ch29-theata-abp', 'ch30-theta-abp']
if low==8:
#px6=arr3
pxt = ['ch25-alpha-abp', 'ch26-alpha-abp', 'ch29-alpha-abp', 'ch30-alpha-abp']
for i, j, k, z in zip (pk1, pk77,pxu,pxt):
freq_res=i[1]-i[0]
idx_d = np.logical_and (i >= low, i <= high)
u1 = simps (j[idx_d], dx=freq_res)
if low==0.5: # delta cal
# u1=simps(j[idx_d], dx=freq_res)
#print(u1)
arr1.append(u1) # delta array
print(z,arr1[k])
if low==4: #// theta cal
# u1 = simps (j[idx_d], dx=freq_res)
arr2.append(u1) # theta array
print(z,arr2[k])
if low==8: #// alpha cal
# u1 = simps (j[idx_d], dx=freq_res)
arr3.append (u1) # alpha arra
print(z,arr3[k])
print('exit out')
print()
print()
#////////////////////////////////////
print('\\\\print the power to each channels ////')
wd4=[0,1,2]
rel1=[] # location to relitive power of each band delta band
delta/delta+heta+alpha 1x4
rel2=[]
rel3=[]
wx1=['ch25-delta-pb','ch26-delta-pb','ch29-delta-pb','ch30-delta-pb']
wx2= ['ch25-theta-pb','ch26-theta-pb','ch29-theta-pb','ch30-theta-pb']
wx3=['ch25-alpha-pb','ch26-aloha-pb','ch29-alpha-pb','ch30-alpha-pb']
wx4=[0,1,2,3]
for h in wd4:

 for i,j,k,g in zip(wx1,wx2,wx3,wx4):
 if h==0:
print(i,arr1[g])
 if h==1:
print(j,arr2[g])
 if h==2:
 print(k,arr3[g])
print('exit 55')
print()
#calculate the reliative total power for ech channel
print(nn2,'total relative power for each ')

wz1=['R_ch25-delta','R-ch26-delta','R-ch29-delta','R-ch30-delta']
wz2=['R_ch25-theta','R-ch26-theta','R-ch29-theta','R-ch30-theta']
wz3=['R_ch25-alpha','R-ch26-alpha','R-ch29-alpha','R-ch30-alpha']
wy=[0,1,2]
wy2=[0,1,2,3]
for y in wy:
 for j,k,m,l in zip(wy2,wz1,wz2,wz3):
 # for i,j,k,m in zip(wz1,wz2,wz3,wx4):
 if y==0:
 av=arr1[j]/(arr1[j]+arr2[j]+arr3[j]) # for delta band for single ch
 av=format(av,'.5f')
 rel1.append(av)
 print(k,rel1[j])
 if y==1:
 av = arr2[j] / (arr1[j]+arr2[j] + arr3[j])

av = format (av, '.5f')
 rel2.append (av)
 print(m,rel2[j])
 if y==2:
 av = arr2[j] / (arr1[j] + arr2[j] + arr3[j])
 av = format (av, '.5f')
 rel3.append (av)
 print(l,rel3[j])
print('out exit exit ')
print()
print()
#/////////////////////////////////////////\\\\\\\\\\\\\\\\\
print(nn2,'cal the hemispher power/2nd way')
rxx=[] # rxx array location / declration to store thw hem /powe 6 value only
wy=[0,1,2]
wy1=[0,1,2,3]
for i in wy:
 for j in wy1:
 if i==0 and j==0:
 ax1=(abs(arr1[j]-arr1[j+1])/(arr1[j]+arr1[j+1]))
 ax2=(abs(arr1[j+2]-arr1[j+3])/(arr1[j+2]+arr1[j+3]))
 vv1 = format (ax1, '.4f')
 vv2 = format (ax2, '.4f')
 rxx.append(vv1)
 rxx.append(vv2)
 print('fir-val',rxx[j])
 print('2nd-val',rxx[j+1])
 print()
 if i==1 and j==0:
 a3 =(abs( arr2[j] - arr2[j + 1]) / (arr2[j] + arr2[j + 1]))
 a4 = (abs(arr2[j + 2] - arr2[j + 3]) / (arr2[j + 2] + arr2[j + 3]))
 vv3 = format (a3, '.4f')
 vv4= format (a4, '.4f')
 rxx.append(vv3)
 rxx.append(vv4)
 print ('3rd-val',rxx[j+2])
 print ('four-val',rxx[j+3])

print()
 if i==2 and j==0:
 a5 = (abs(arr3[j] - arr3[j + 1]) / (arr3[j] + arr3[j + 1]))
 a6 = (abs(arr3[j + 2] - arr3[j + 3]) / (arr3[j + 2] + arr3[j + 3]))
 vv5 = format (a5, '.4f')
 vv6= format (a6, '.4f')
 rxx.append(vv5)
 rxx.append(vv6)
 print ('fif-val', rxx[j+4])
 print ('six-val', rxx[j+5])
print('exit 66')

datax=(harr+staa+rel1+rel2+rel3+rxx)
print('the data',datax)
print()
arr = np.concatenate((harr,staa,rel1,rel2,rel3,rxx))
#print('the new',arr)
# #print(datax.shape)
#np.savetxt('F:\my plot_data/riyad600.csv', arr, delimiter=",")
#np.savetxt('F:\my plot_data\dataset/pr1.csv', datax)
pd.DataFrame(datax).to_csv('F:\my plot_data/pr1.csv')
# print('999')

arr.tofile('F:\my plot_data/pr2.csv',sep=',') #using 2 nd way using tofile creat pr2/
print('using datafram way pr1',nn2)
with open("F:\my plot_data\pr1.csv", 'r') as file: # read the file
csvreader = csv.reader(file, delimiter=':')
#csvreader = csv.reader (file)
for row in csvreader:
print(row)
print()
print('using to.file way pr2',nn2)
with open("F:\my plot_data\pr2.csv", 'r') as file: # read the file
csvreader = csv.reader(file, delimiter=':')
#csvreader = csv.reader (file)
for row in csvreader:
```

```
print(row)
```

Python program list ***hurst.py*** to compute the Hurst function attributes in chapter 3.

```
import Numpy as np

 matplotlib.pyplot as plt

from hurst import compute_Hc, random_walk

# series using the import random_walk() function.

# Use random_walk() function to generate a random walk series

s = random_walk(10000) # Parameter passed must be 100 or above. Next, we
calculate the Hurst exponent #(H) and the (e))

 # compute returns a tuple of 3 values

H, c, val = compute_Hc(s)
```

The python program list **welch.py** to compute the power density by welch method that refer with **chapter three** and display the output in figure 3.7.

APPANDIX -A --Feature Extraction Program --- Source Code -- Python List

```
import numpy as np
from scipy import signal
import matplotlib.pyplot as plt
rng = np.random.default_rng()
/#Generate a test signal, a 2 Vrms sine wave at 1234 Hz, corrupted
by 0.001
# V**2/Hz of white noise sampled at 10 kHz.
fs = 10e3
N = 1e5
```

```
amp = 2*np.sqrt(2)
freq = 1234.0
noise_power = 0.001 * fs / 2
time = np.arange(N) / fs
x = amp*np.sin(2*np.pi*freq*time)
x += rng.normal(scale=np.sqrt(noise_power), size=time.shape)
Compute and plot the power spectral density.
f, Pxx_den = signal.welch(x, fs, nperseg=1024)
plt.semilogy(f, Pxx_den)
  plt.ylim([0.5e-3, 1])
  plt.xlabel('frequency [Hz]')
  plt.ylabel('PSD [V**2/Hz]')
  plt.show()
```

APPANDIX -A --Feature Extraction Program --- Source Code -- Python List

pcc.py python program that refer to in chapter 3, to compute the power band, the output shown in figure 3.8.

```
from matplotlib import pyplot as plt # import the library
from scipy import signal
import numpy as np
data = np.loadtxt('data.txt')
import seaborn as sns
sns.set(font_scale=1.2)# function that used with the powar band
sf=100# sampling frequency
 win=4*sf # Define window length (4 seconds)
freqs, psd = signal.welch(data, sf, nperseg=win) # read frequency component with power for certain time
sns.set(font_scale=1.2, style='white')
plt.figure(figsize=(8, 6))
plt.plot(freqs, psd, color='k', lw=2)
```

```
plt.xlabel('Frequency (Hz)')
plt.ylabel('Power spectral density (V^2 / Hz)')
plt.ylim([0, psd.max() * 1.1])
plt.title("Welch's periodogram")
plt.xlim([0, freqs.max()])
sns.despine()
plt.show()
```

Appendix (B)

Data Appendix That Contains Forty Participants Arranged with Four Table

Appendix-B

IN THAT APPENDIX-B (DATA Appendix) arrange all attributes within four tables each table contain 10 participants.

The attribute contains 26 items for each participant

1. **hurst exponent includes four values representing four channels**
2. **The mean attribute represents four values that refer to four channels.**
3. **Relative Power Band ratios in each channel include 3 band (α, β, δ) for low frequency. So, each channel contains 3 bands of frequency, and all attribute is 12 values**
4. **Relative Band Power of hemisphere refers to the channels position in the front and back of the head contain 6 values.**
5. **all values represent the matrix containing 26x40 so its includes 1040 values.**
6. **tabel B-1-a, table B-1-b, table B-1-c, table B-1-d.**

Table B-1-A contain participants group-1 from (1 to 10)

Table B-1-A	**contain participants group-1 from (1 to 10)**									
	Par-1	**Par-2**	**Par-3**	**Par-4**	**Par-5**	**Par-6**	**Par-7**	**Par-8**	**Par-9**	**Par-10**
1/hurst-1	**0.2127**	**0.2376**	**0.2427**	**0.2376**	**0.2276**	**0.2727**	**0.2427**	**0.2376**	**0.2427**	**0.2437**
2/hurst-2	**0.2727**	**0.3127**	**0.1727**	**0.2276**	**0.3276**	**0.2327**	**0.1727**	**0.2176**	**0.2727**	**0.3127**
3/hurst-3	**0.2627**	**0.2176**	**0.4327**	**0.2273**	**0.3276**	**0.2627**	**0.3627**	**0.2276**	**0.3627**	**0.2627**
4/hurst-4	**0.2627**	**0.2276**	**0.3627**	**0.2276**	**0.2276**	**0.2627**	**0.2627**	**0.3276**	**0.2627**	**0.2627**
5/mean1	**1.2120**	**3.1490**	**1.2120**	**3.14907**	**3.14907**	**1.2120**	**1.2120**	**3.1490**	**1.2120**	**1.2120**
6/mean2	**2.4408**	**-3.536**	**2.4408**	**-3.5364**	**3.53642**	**2.4408**	**2.4408**	**3.5364**	**2.4408**	**2.4408**
7/mean3	**1.1891**	**2.0482**	**1.1891**	**-2.0482**	**-2.0482**	**1.1891**	**1.1891**	**-2.048**	**1.1891**	**1.1891**
8/mean4	**1.2334**	**3.1490**	**1.2334**	**3.14907**	**3.14907**	**1.2334**	**1.2334**	**3.1490**	**1.2334**	**1.2334**
9/Rprr1	**0.2376**	**0.1553**	**0.2376**	**0.1553**	**0.1553**	**0.2376**	**0.2376**	**0.1553**	**0.2376**	**0.2376**
10/Rpr2	**0.2276**	**0.1559**	**0.2276**	**0.1559**	**0.1559**	**0.2276**	**0.2276**	**0.1559**	**0.2276**	**0.2276**
11/Rpr3	**0.2276**	**0.1553**	**0.2276**	**0.1553**	**0.1553**	**0.2276**	**0.2276**	**0.1553**	**0.2276**	**0.2276**
12/Rpr4	**3.1490**	**0.1539**	**3.1490**	**0.1539**	**0.1539**	**3.1490**	**3.1490**	**0.1539**	**3.1490**	**3.1490**
13/Rpr5	**3.5364**	**0.1013**	**3.5364**	**0.1013**	**0.1013**	**3.5364**	**3.5364**	**0.1013**	**3.5364**	**3.5364**
14/Rpr6	**2.0482**	**0.0940**	**2.0482**	**0.0940**	**0.0940**	**2.0482**	**2.0482**	**0.0940**	**2.0482**	**2.0482**

Appendix -B – Data Appendix That Contains Forty Participants Arranged with Four Table

Table B-1-A	**contain participants group-1 from (1 to 10)**									
15/Rpr7	**0.2276**	**0.0903**	**0.2276**	**0.0903**	**0.0903**	**0.2276**	**0.2276**	**0.0903**	**0.2276**	**0.2276**
16/Rpr8	**0.1325**	**0.2553**	**0.1345**	**0.1583**	**0.1555**	**0.1305**	**0.1345**	**0.1553**	**0.1360**	**0.1345**
17/Rpr9	**0.1091**	**0.1843**	**0.1091**	**0.1844**	**0.1847**	**0.1092**	**0.1291**	**0.1843**	**0.1093**	**0.1091**
18/Rpr10	**0.1296**	**0.1013**	**0.1293**	**0.1013**	**0.1113**	**0.1292**	**0.1293**	**0.1013**	**0.1297**	**0.1292**
19/Rpr11	**0.1335**	**0.1759**	**0.1355**	**0.1749**	**0.1719**	**0.1155**	**0.1355**	**0.1759**	**0.2355**	**0.1355**
20/par12	**0.2345**	**0.1553**	**0.1345**	**0.1453**	**0.1553**	**0.1335**	**0.1325**	**0.1553**	**0.1345**	**0.2345**
21/Rprh1	**0.271**	**0.102**	**0.219**	**0.122**	**0.102**	**0.179**	**0.379**	**0.102**	**0.179**	**0.279**
22/Rprh2	**0.074**	**0.206**	**0.034**	**0.201**	**0.206**	**0.044**	**0.074**	**0.206**	**0.034**	**0.074**
23/ prh3	**0.032**	**0.444**	**0.032**	**0.442**	**0.444**	**0.032**	**0.037**	**0.244**	**0.034**	**0.032**
24/Rprh4	**0.047**	**0.083**	**0.047**	**0.068**	**0.078**	**0.051**	**0.057**	**0.084**	**0.051**	**0.057**
25/ prh5	**0.054**	**0.043**	**0.053**	**0.046**	**0.043**	**0.055**	**0.053**	**0.047**	**0.033**	**0.052**
26/Rprh6	**0.017**	**0.003**	**0.007**	**0.043**	**0.042**	**0.009**	**0.007**	**0.042**	**0.007**	**0.055**

Table B-1-B contain participants group-2 from (11 to 20)

Table B-1-B	**Contain 10 participants group- 2 from (11-20)**									
	Par11	**Par12**	**Par13**	**Par14**	**Par15**	**Par16**	**Par17**	**Par18**	**Par19**	**Par20**
1/hurst-1	**0.2427**	**0.2376**	**0.2427**	**0.2376**	**0.2376**	**0.2427**	**0.2427**	**0.2376**	**0.2427**	**0.2427**
2/hurst-2	**0.2727**	**0.2176**	**0.2427**	**0.2764**	**0.3276**	**0.1727**	**0.2737**	**0.1276**	**0.2727**	**0.2727**
3/hurst-3	**0.2627**	**0.2176**	**0.2637**	**0.2273**	**0.2276**	**0.2727**	**0.2627**	**0.2476**	**0.3627**	**0.2637**
4/hurst-4	**0.2627**	**0.2276**	**0.2627**	**0.2476**	**0.1276**	**0.2627**	**0.1627**	**0.2276**	**0.2637**	**0.2627**
5/mean1	**1.2120**	**3.1290**	**1.2120**	**3.1190**	**3.1390**	**1.2320**	**1.2120**	**3.1490**	**1.2120**	**1.2120**
6/mean2	**2.4408**	**-3.536**	**2.4408**	**-3.5364**	**3.53642**	**2.4438**	**2.4108**	**3.5364**	**2.4418**	**2.4458**
7/mean3	**1.1891**	**2.0482**	**1.1891**	**-2.0482**	**-2.0482**	**1.1791**	**1.1891**	**-2.048**	**1.1891**	**1.1891**
8/mean	**1.2334**	**3.1490**	**1.2334**	**3.1490**	**3.1490**	**1.2334**	**1.2334**	**3.1490**	**1.2334**	**1.2334**
9/Rprr1	**0.2376**	**0.1553**	**0.2376**	**0.1553**	**0.1553**	**0.2376**	**0.2376**	**0.1553**	**0.2376**	**0.2376**
10/Rpr2	**0.2276**	**0.1559**	**0.2276**	**0.1559**	**0.1559**	**0.2276**	**0.2276**	**0.1559**	**0.2276**	**0.2276**
11/Rpr3	**0.2276**	**0.1553**	**0.2276**	**0.1553**	**0.1553**	**0.2276**	**0.2276**	**0.1553**	**0.2276**	**0.2276**
12/Rpr4	**3.1490**	**0.1539**	**3.1490**	**0.1539**	**0.1539**	**3.1490**	**3.1490**	**0.1539**	**3.1490**	**3.1490**
13/Rpr5	**3.5364**	**0.1013**	**3.5364**	**0.1013**	**0.1013**	**3.5364**	**3.5364**	**0.1013**	**3.5364**	**3.5364**
14/Rpr6	**2.0482**	**0.0940**	**2.0482**	**0.0940**	**0.0940**	**2.0482**	**2.0482**	**0.0940**	**2.0482**	**2.0482**

Appendix -B – Data Appendix That Contains Forty Participants Arranged with Four Table

Table B-1-B	**Contain 10 participants group- 2 from (11-20)**									
15/Rpr7	**0.2276**	**0.0903**	**0.2276**	**0.0903**	**0.0903**	**0.2276**	**0.2276**	**0.0903**	**0.2276**	**0.2276**
16/Rpr8	**0.1345**	**0.1553**	**0.1345**	**0.1553**	**0.1553**	**0.1345**	**0.1345**	**0.1553**	**0.1345**	**0.1345**
17/Rpr9	**0.1091**	**0.1843**	**0.1091**	**0.1843**	**0.1843**	**0.1091**	**0.1091**	**0.1843**	**0.1091**	**0.1091**
18/Rpr10	**0.1292**	**0.1013**	**0.1292**	**0.1013**	**0.1013**	**0.1292**	**0.1292**	**0.1013**	**0.1292**	**0.1292**
19/Rpr11	**0.1355**	**0.1759**	**0.1355**	**0.1759**	**0.1759**	**0.1355**	**0.1355**	**0.1759**	**0.1355**	**0.1355**
20/par12	**0.1345**	**0.1553**	**0.1345**	**0.1553**	**0.1553**	**0.1345**	**0.1345**	**0.1553**	**0.1345**	**0.1345**
21/Rprh1	**0.279**	**0.102**	**0.279**	**0.102**	**0.102**	**0.279**	**0.279**	**0.102**	**0.279**	**0.279**
22/Rprh2	**0.074**	**0.206**	**0.074**	**0.206**	**0.206**	**0.074**	**0.074**	**0.206**	**0.074**	**0.074**
23/Rprh3	**0.032**	**0.444**	**0.032**	**0.444**	**0.444**	**0.032**	**0.032**	**0.444**	**0.032**	**0.032**
24/ prh4	**0.057**	**0.088**	**0.057**	**0.088**	**0.088**	**0.057**	**0.057**	**0.088**	**0.057**	**0.057**
25/Rprh5	**0.053**	**0.043**	**0.053**	**0.043**	**0.043**	**0.053**	**0.053**	**0.043**	**0.053**	**0.053**
26/ prh6	**0.007**	**0.003**	**0.007**	**0.043**	**0.043**	**0.007**	**0.007**	**0.042**	**0.007**	**0.02**

Table B-1-C contain participants group-3 from (21 to 30)

Table B-1-C	**Contain 10 participants group 3 from (21-30)**									
	Par21	**Par22**	**Par23**	**Par24**	**Par25**	**Par26**	**Par27**	**Par28**	**Par29**	**Par30**
1/hurst-1	**0.2427**	**0.2376**	**0.2427**	**0.2376**	**0.2376**	**0.2427**	**0.2427**	**0.2376**	**0.2427**	**0.2427**
2/hurst-2	**0.2727**	**0.2276**	**0.2727**	**0.2276**	**0.2276**	**0.2727**	**0.2727**	**0.2276**	**0.2727**	**0.2727**
3/hurst-3	**0.2627**	**0.2176**	**0.2627**	**0.2273**	**0.2276**	**0.2627**	**0.2627**	**0.2276**	**0.2627**	**0.2627**
4/hurst-4	**0.2627**	**0.2376**	**0.2627**	**0.2376**	**0.2276**	**0.2627**	**0.2827**	**0.2276**	**0.2617**	**0.2627**
5/mean1	**1.2120**	**3.1490**	**1.2120**	**3.1490**	**3.1190**	**1.2120**	**1.2120**	**3.1490**	**1.2220**	**1.2120**
6/mean2	**2.4408**	**-3.536**	**2.4408**	**-3.536**	**3.564**	**2.4418**	**2.4608**	**3.5364**	**2.4308**	**2.4408**
7/mean3	**1.1891**	**2.0482**	**1.1891**	**-2.048**	**-2.048**	**1.1891**	**1.1791**	**-2.048**	**1.1891**	**1.1831**
8/mean4	**1.2334**	**3.1490**	**1.2134**	**3.1490**	**3.1190**	**1.2334**	**1.2334**	**3.1690**	**1.5334**	**1.2334**
9/Rprr1	**0.2376**	**0.1553**	**0.2376**	**0.1553**	**0.2553**	**0.2176**	**0.2476**	**0.1553**	**0.2376**	**0.2476**
10/Rpr2	**0.2276**	**0.1559**	**0.2286**	**0.1559**	**0.1559**	**0.2276**	**0.2276**	**0.1559**	**0.2276**	**0.2276**
11/Rpr3	**0.2176**	**0.1553**	**0.2226**	**0.1553**	**0.1553**	**0.2176**	**0.2273**	**0.1553**	**0.2416**	**0.2376**
12/Rpr4	**3.1390**	**0.1291**	**3.1490**	**0.1339**	**0.1539**	**3.1490**	**3.2490**	**0.1539**	**3.1490**	**3.1690**
13/Rpr5	**3.5364**	**0.1013**	**3.4164**	**0.1013**	**0.1013**	**3.5364**	**3.5334**	**0.1013**	**3.2364**	**3.5364**
14/Rpr6	**2.0482**	**0.0940**	**2.0482**	**0.0940**	**0.0940**	**2.0482**	**2.0482**	**0.0940**	**2.0482**	**2.0482**

Appendix -B – Data Appendix That Contains Forty Participants Arranged with Four Table

Table B-1-C	**Contain 10 participants group 3 from (21-30)**									
15/Rpr7	**0.2476**	**0.0912**	**0.1276**	**0.0903**	**0.0610**	**0.2276**	**0.2476**	**0.0903**	**0.2273**	**0.2276**
16/Rpr8	**0.1345**	**0.1553**	**0.1345**	**0.1553**	**0.1553**	**0.1345**	**0.1345**	**0.1553**	**0.1345**	**0.1345**
17/Rpr9	**0.1091**	**0.1843**	**0.1091**	**0.1843**	**0.1843**	**0.1091**	**0.1091**	**0.1843**	**0.1091**	**0.1091**
18/Rpr10	**0.1292**	**0.1013**	**0.1292**	**0.1013**	**0.1023**	**0.1291**	**0.1292**	**0.1013**	**0.1392**	**0.1292**
19/Rpr11	**0.1355**	**0.1759**	**0.1355**	**0.1759**	**0.1729**	**0.1255**	**0.1655**	**0.1759**	**0.1315**	**0.1355**
20/par12	**0.1315**	**0.1553**	**0.1325**	**0.1653**	**0.1553**	**0.1145**	**0.1345**	**0.1553**	**0.1245**	**0.1345**
21/Rprh1	**0.279**	**0.102**	**0.219**	**0.112**	**0.101**	**0.279**	**0.209**	**0.102**	**0.229**	**0.279**
22/Rprh2	**0.073**	**0.216**	**0.077**	**0.206**	**0.216**	**0.075**	**0.074**	**0.206**	**0.072**	**0.074**
23/Rprh3	**0.036**	**0.441**	**0.032**	**0.447**	**0.444**	**0.032**	**0.042**	**0.444**	**0.035**	**0.032**
24/ prh4	**0.052**	**0.084**	**0.054**	**0.082**	**0.088**	**0.057**	**0.051**	**0.081**	**0.052**	**0.057**
25/Rprh5	**0.053**	**0.043**	**0.053**	**0.043**	**0.043**	**0.053**	**0.053**	**0.043**	**0.053**	**0.053**
26/ prh6	**0.003**	**0.003**	**0.017**	**0.042**	**0.043**	**0.007**	**0.004**	**0.042**	**0.007**	**0.02**

Table B-1-D contain participants group-4 from (31 to 40)

Table B-1-D	**Contain 10 participants group 4 from (31-40)**									
	Par31	**Par32**	**Par33**	**Par34**	**Par35**	**Par36**	**Par37**	**Par38**	**Par39**	**Par40**
1/hurst1	**0.2327**	**0.2176**	**0.2427**	**0.2476**	**0.2376**	**0.2427**	**0.2417**	**0.2376**	**0.2427**	**0.2427**
2/hurst2	**0.3727**	**0.2276**	**0.2727**	**0.2576**	**0.2276**	**0.2727**	**0.2727**	**0.2276**	**0.1727**	**0.2227**
3/hurst3	**0.2327**	**0.2176**	**0.2627**	**0.3273**	**0.2276**	**0.2637**	**0.2627**	**0.2176**	**0.2627**	**0.2627**
4/hurst4	**0.2637**	**0.2176**	**0.2627**	**0.2276**	**0.2216**	**0.2627**	**0.2227**	**0.3276**	**0.2127**	**0.2627**
5/mean11	**1.2120**	**3.2490**	**1.2120**	**3.1490**	**3.1490**	**1.2120**	**2.2120**	**3.1490**	**1.2720**	**1.1120**
6/mean 2	**2.3408**	**-3.536**	**2.4408**	**-3.536**	**3.5364**	**2.4408**	**2.4408**	**2.5364**	**2.4408**	**1.4408**
7/mean 3	**1.1591**	**2.0482**	**1.1891**	**-2.048**	**-2.048**	**1.1891**	**1.1891**	**-2.048**	**1.1891**	**1.1891**
8/mean 4	**1.2334**	**3.1490**	**1.1334**	**3.1390**	**3.1490**	**1.2334**	**1.3334**	**3.1390**	**1.2334**	**1.6334**
9/Rprr1	**0.2376**	**0.1553**	**0.2376**	**0.1552**	**0.1553**	**0.2376**	**0.2376**	**0.1153**	**0.2176**	**0.2376**
10/Rpr2	**0.2276**	**0.1559**	**0.2176**	**0.3559**	**0.1559**	**0.2276**	**0.2276**	**0.1559**	**0.2276**	**0.2276**
11/Rpr3	**0.2276**	**0.1553**	**0.2276**	**0.1553**	**0.1553**	**0.2276**	**0.2276**	**0.1553**	**0.2276**	**0.2276**
12/Rpr4	**3.1490**	**0.1539**	**3.1490**	**0.1539**	**0.1539**	**3.1490**	**3.1490**	**0.1539**	**1.1490**	**3.1490**
13/Rpr5	**1.5364**	**0.1013**	**2.5364**	**0.1013**	**0.1013**	**0.5364**	**1.5364**	**0.1013**	**1.2364**	**1.5164**
14/Rpr6	**2.0482**	**0.0940**	**2.0442**	**0.0940**	**0.0940**	**2.0482**	**1.0482**	**0.0940**	**2.0482**	**2.0482**

Appendix -B – Data Appendix That Contains Forty Participants Arranged with Four Table

Table B-1-D	**Contain 10 participants group 4 from (31-40)**									
15/Rpr7	**0.2276**	**0.0913**	**0.2776**	**0.0913**	**0.0903**	**0.2376**	**0.2276**	**0.0903**	**0.2276**	**0.2276**
16/Rpr8	**0.1345**	**0.1553**	**0.1325**	**0.1553**	**0.1253**	**0.1345**	**0.1145**	**0.1553**	**0.1345**	**0.1345**
17/Rpr9	**0.1091**	**0.1843**	**0.1091**	**0.1843**	**0.1843**	**0.1091**	**0.1091**	**0.1843**	**0.1091**	**0.1091**
18/Rpr10	**0.1262**	**0.1023**	**0.1292**	**0.1113**	**0.1013**	**0.1392**	**0.1212**	**0.1013**	**0.1292**	**0.1292**
19/Rpr11	**0.1355**	**0.1759**	**0.1355**	**0.1759**	**0.1359**	**0.1355**	**0.1555**	**0.1759**	**0.1355**	**0.1355**
20/par12	**0.1345**	**0.1553**	**0.1345**	**0.1553**	**0.1553**	**0.1545**	**0.1345**	**0.1553**	**0.1245**	**0.1345**
21/Rprh1	**0.129**	**0.112**	**0.275**	**0.102**	**0.102**	**0.279**	**0.271**	**0.132**	**0.279**	**0.179**
22/Rprh2	**0.071**	**0.206**	**0.074**	**0.211**	**0.202**	**0.034**	**0.074**	**0.206**	**0.024**	**0.072**
23/ Rprh3	**0.032**	**0.144**	**0.032**	**0.414**	**0.444**	**0.032**	**0.032**	**0.444**	**0.032**	**0.032**
24/ Rprh4	**0.057**	**0.088**	**0.057**	**0.088**	**0.088**	**0.057**	**0.057**	**0.088**	**0.057**	**0.057**
25/ Rprh5	**0.053**	**0.041**	**0.052**	**0.043**	**0.043**	**0.053**	**0.053**	**0.043**	**0.053**	**0.053**
26/ Rprh6	**0.007**	**0.003**	**0.007**	**0.043**	**0.043**	**0.007**	**0.007**	**0.042**	**0.007**	**0.022**

Appendix (C)

Contains 4 Participants With 26 Attributes for Each

Appendix-C

More Details (Plots)

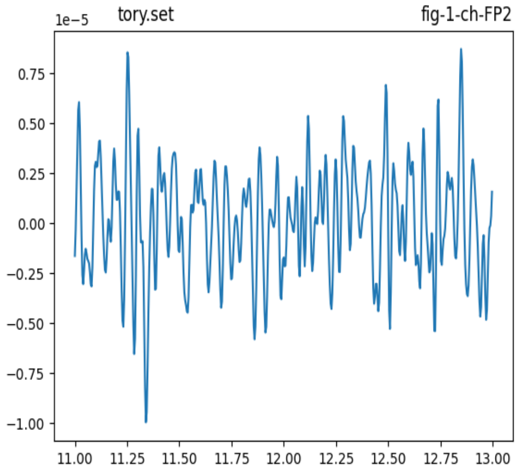
That appendix C contains four participants each participant include 26 attributes as follows

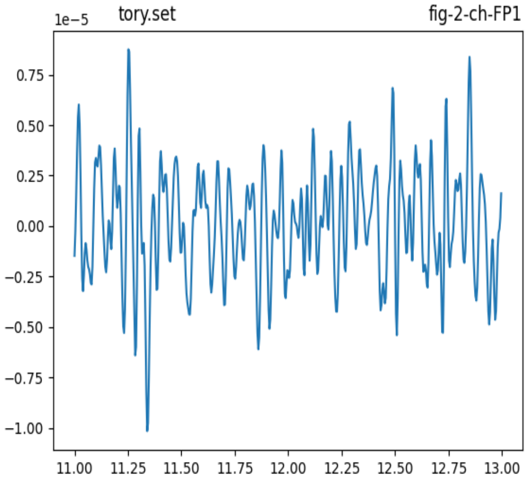
- **Participant code (tory.set), refer to (par1 refer to participant one).**
- **Participant code(sub009.set), refer to par 2.**
- **Participant code (ditory.set), refer to par 3.**
- **Participant code (sub013.set, refer to par 4**
- **For all participants share with**
- **FIG 1,2,3,4 ---- represent the time series of each channels E30, E29,E26,E25-sensore names, four plots**
- **FIG 6,7,8,9 --- Represent the power spectrum plot, using the Welch method, for a long time, Four plot.**
- **Fig 10,11,12,13,14, --- represent the power spectrum plot, using welch, for fixed window 4 sec, five plots.**
- **Fig 15,16, ------26 represent the band-power Delta, Theta, Alpha, with the power density by welch way, 12 plot.**

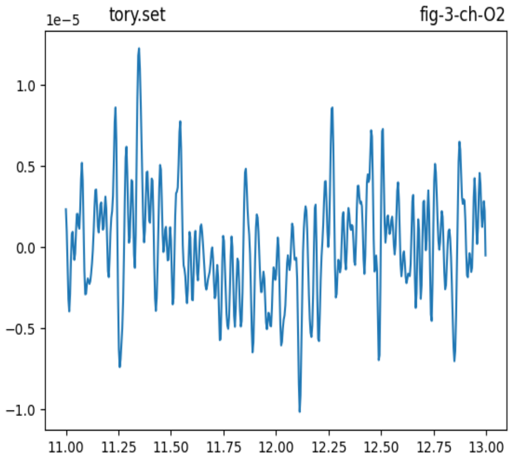
- **All below figures arrange under name Figure C-1 Features Attributes**

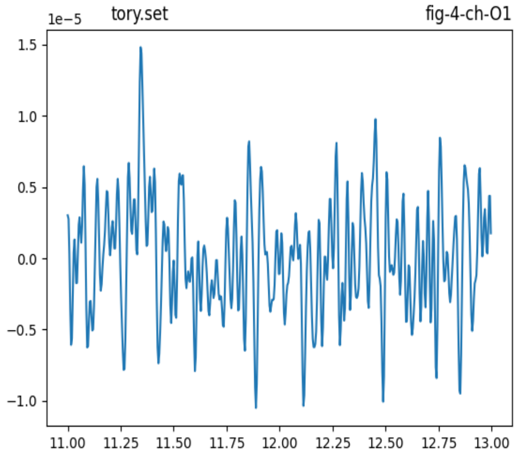
Appendix –C Contains 4 Participants With 26 Attributes for Each

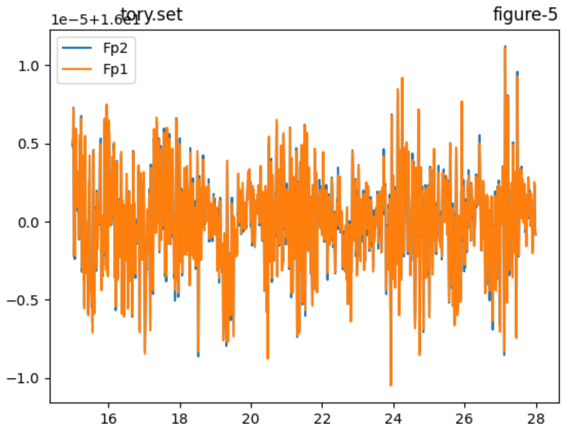
Participant number one (tory.set)

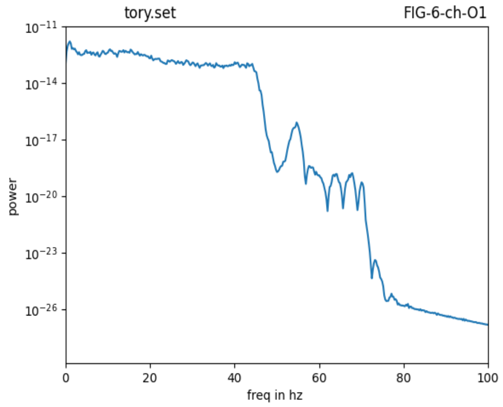






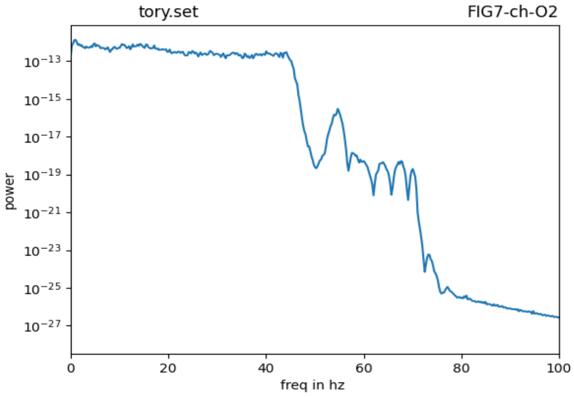


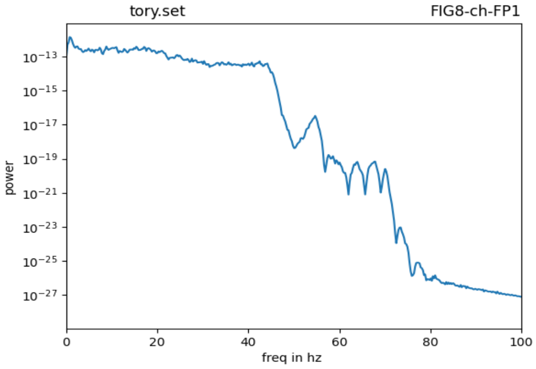


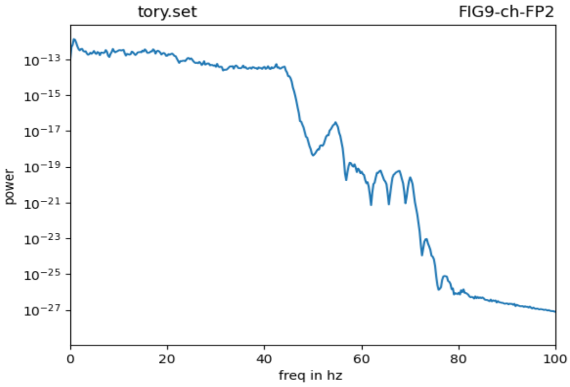


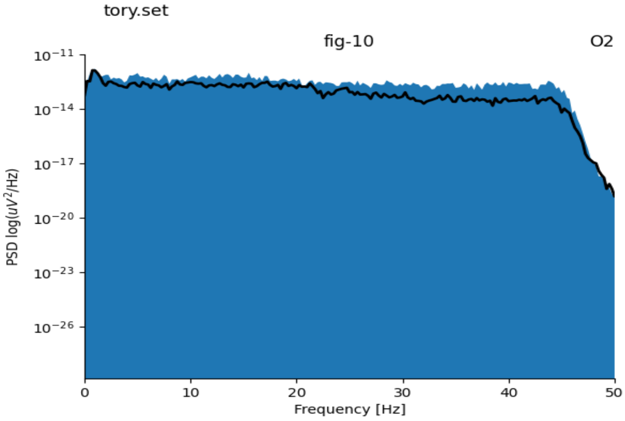
Appendix –C Contains 4 Participants With 26 Attributes for Each

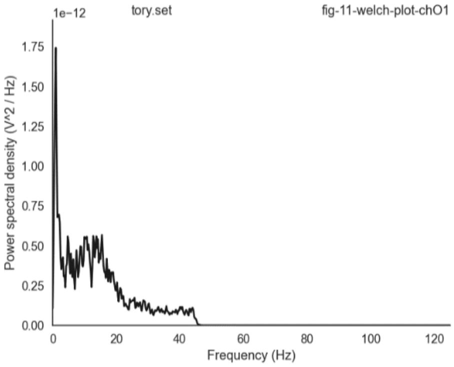
Participant number one (tory.set (tory.set)

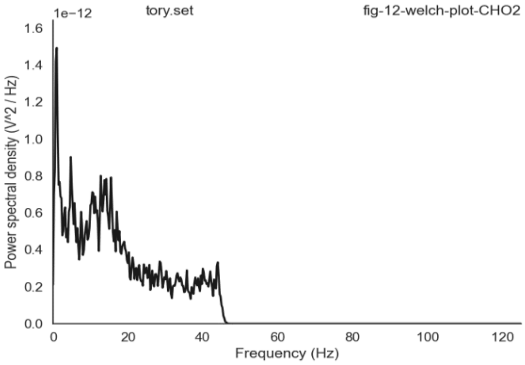






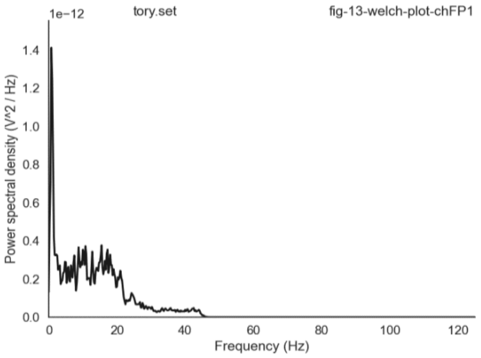


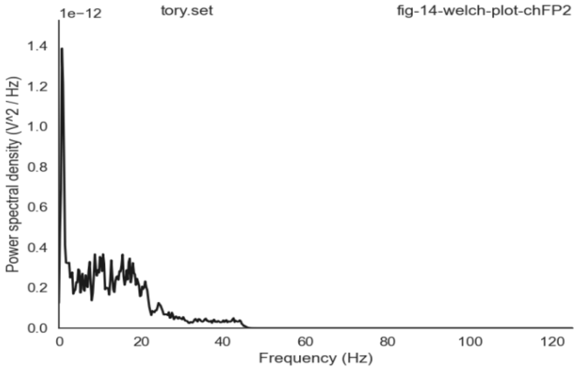


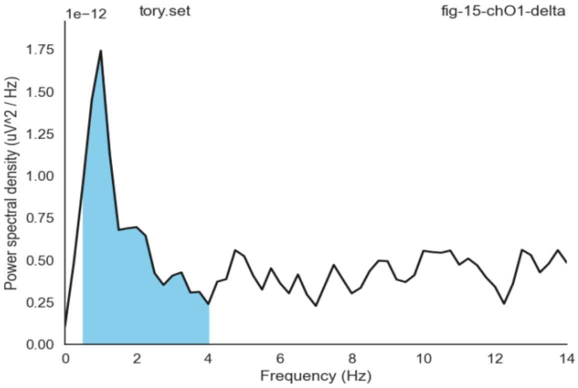


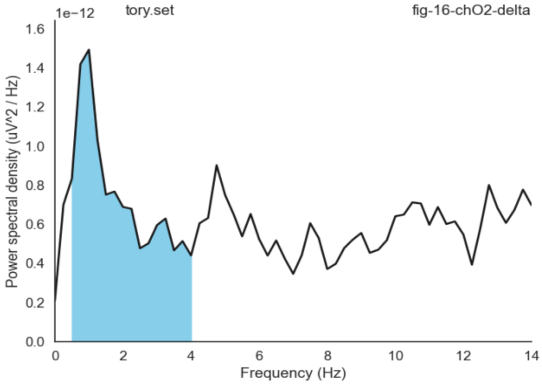
Appendix –C Contains 4 Participants With 26 Attributes for Each

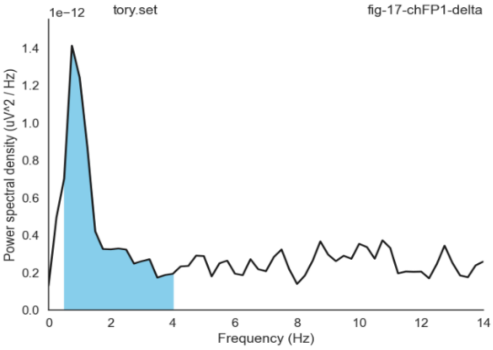
participant number one (tory.set)

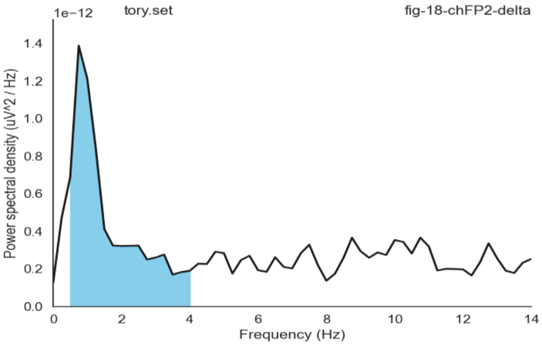






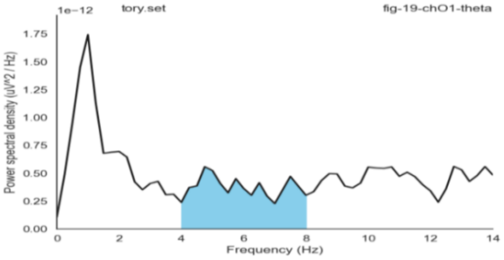


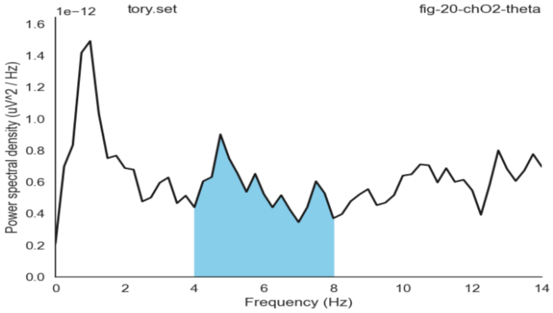


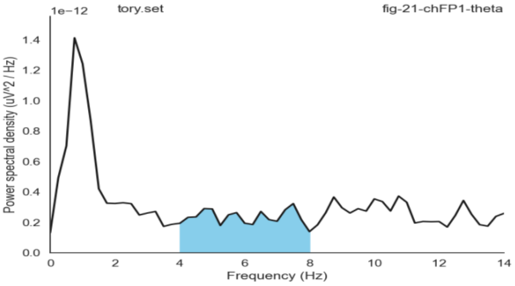


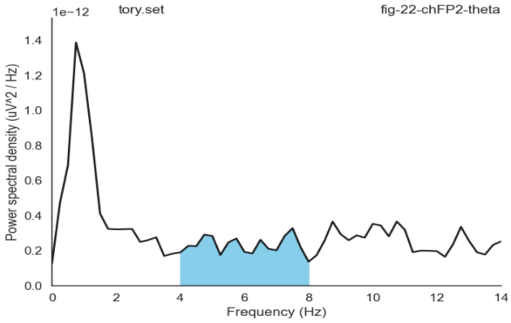
Appendix –C Contains 4 Participants With 26 Attributes for Each

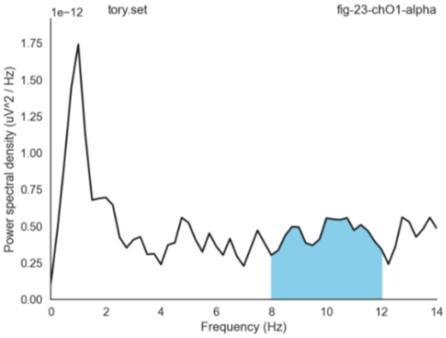
participant number one (tory.set)

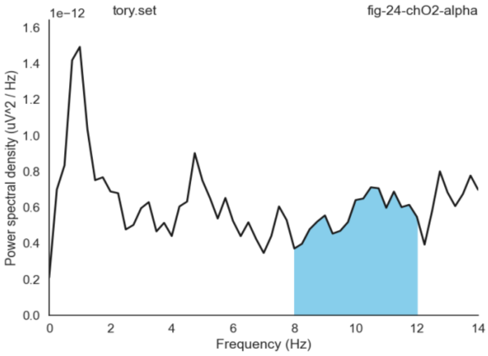






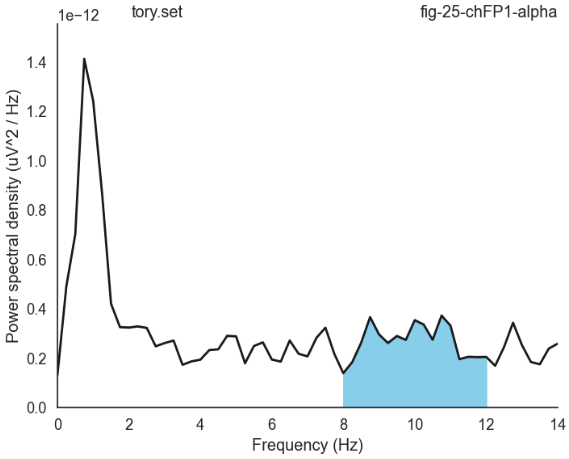


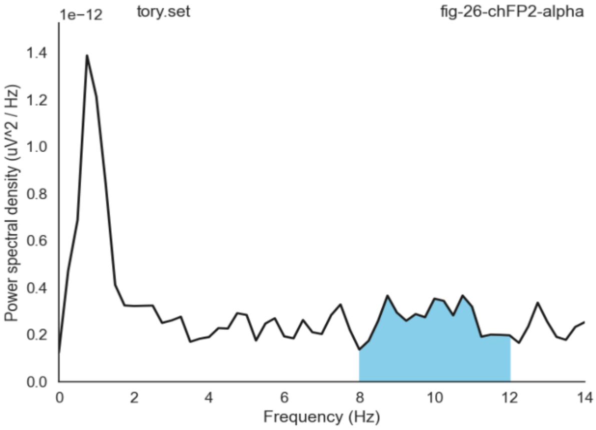




Appendix –C Contains 4 Participants With 26 Attributes for Each

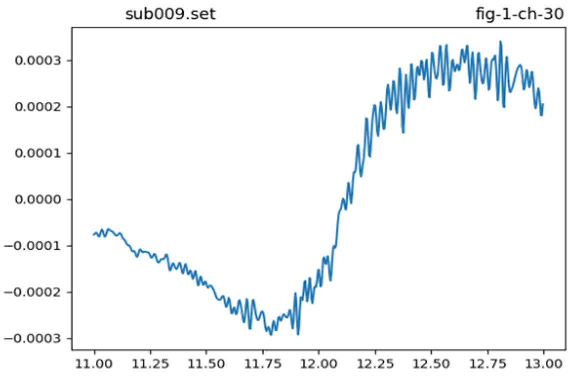
participant number one (tory.set)

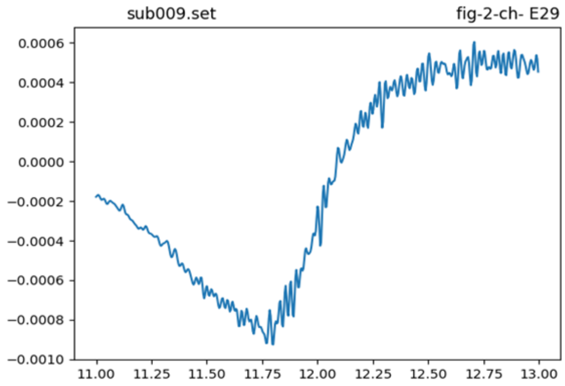


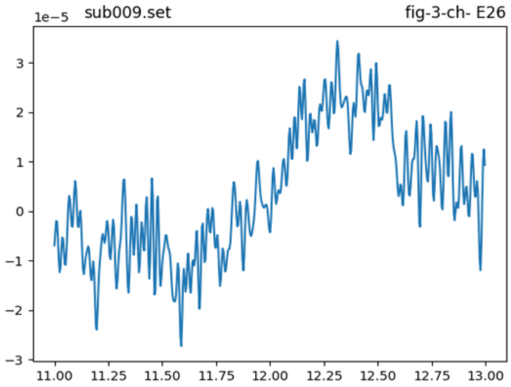


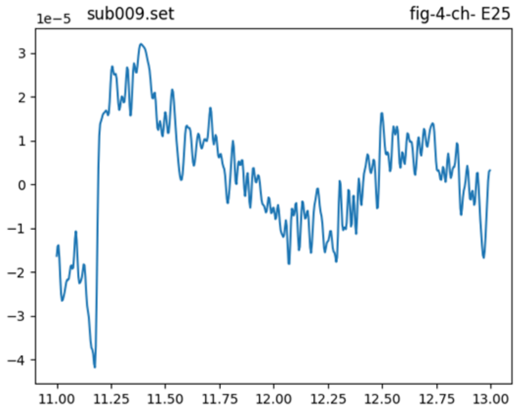
Appendix –C Contains 4 Participants With 26 Attributes for Each

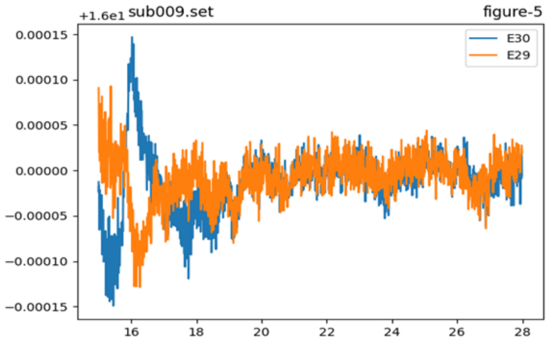
participant number two (sub009.st)

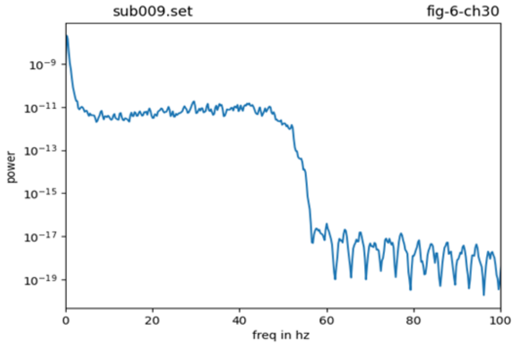

sub009.set fig-1-ch-30


sub009.set fig-2-ch- E29


sub009.set fig-3-ch- E26

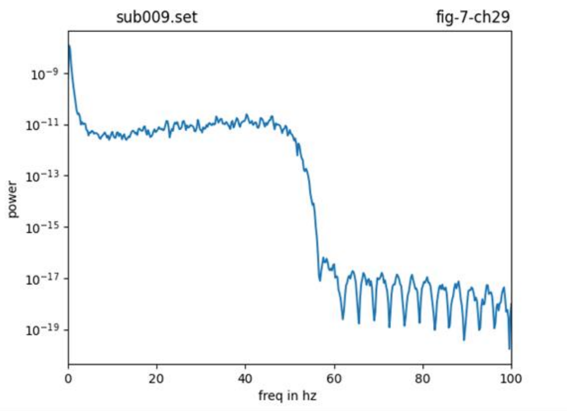

sub009.set fig-4-ch- E25

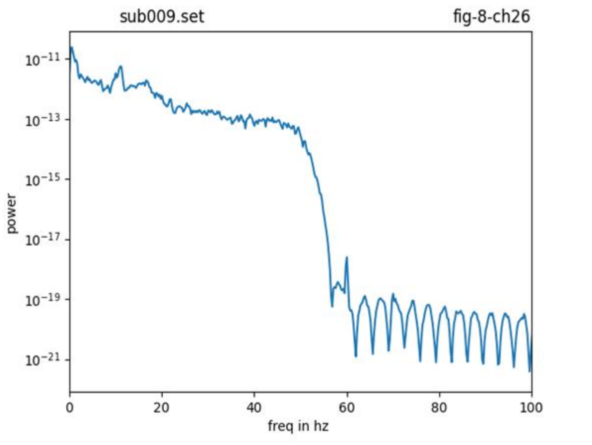

sub009.set figure-5

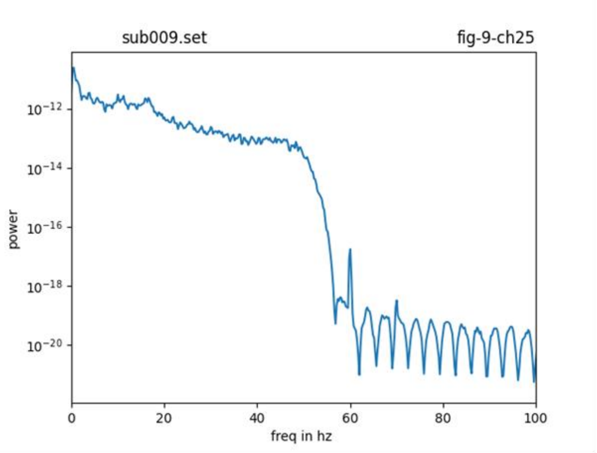

sub009.set fig-6-ch30

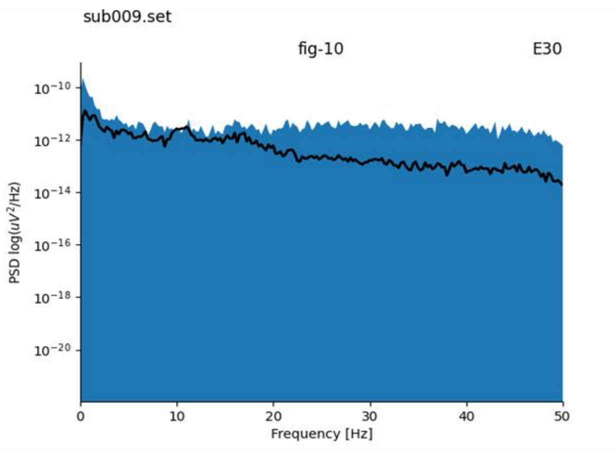
Appendix –C Contains 4 Participants With 26 Attributes for Each

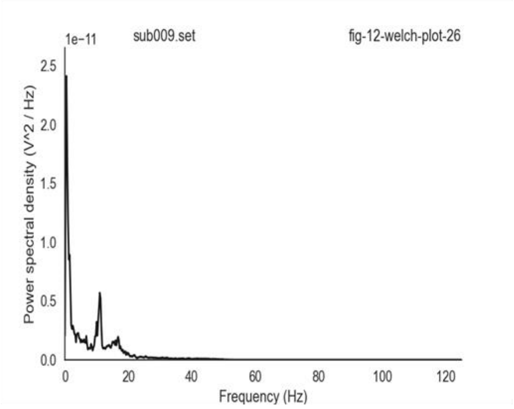
participant number two (sub009.set)

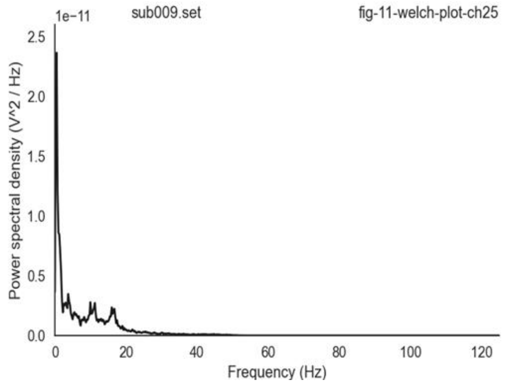






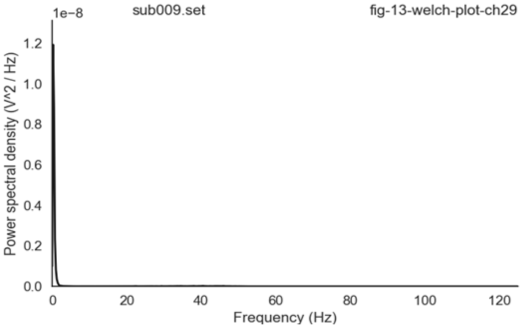


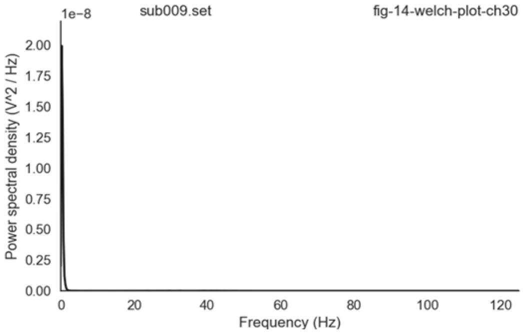


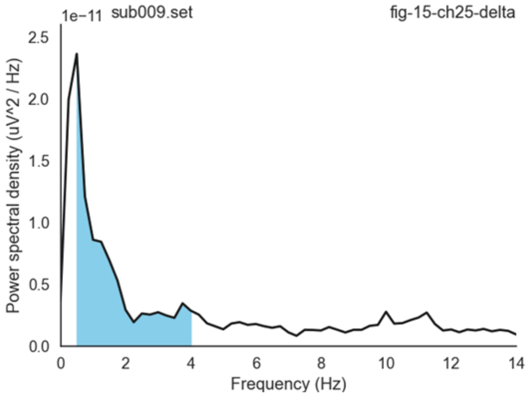


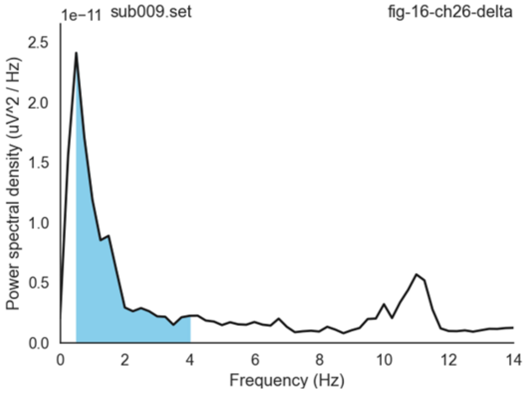
Appendix –C Contains 4 Participants With 26 Attributes for Each

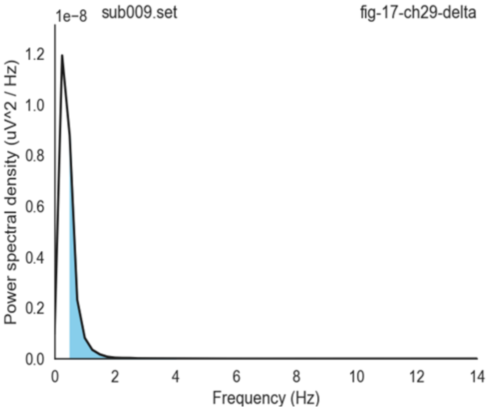
participant number two (sub009.set)

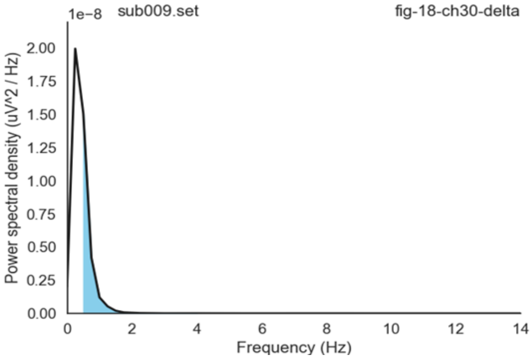






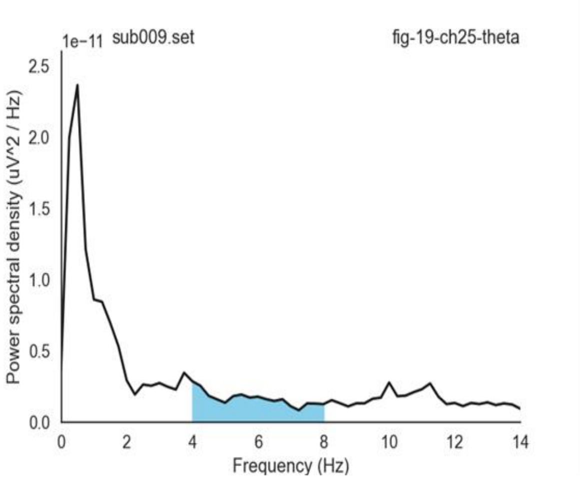


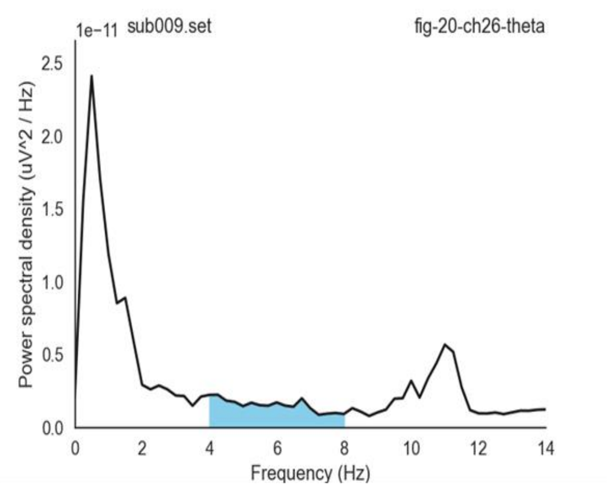


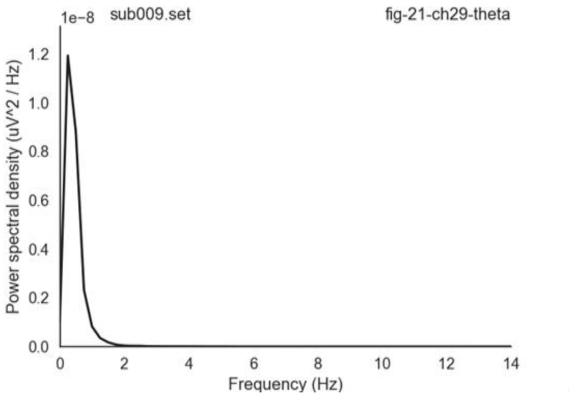


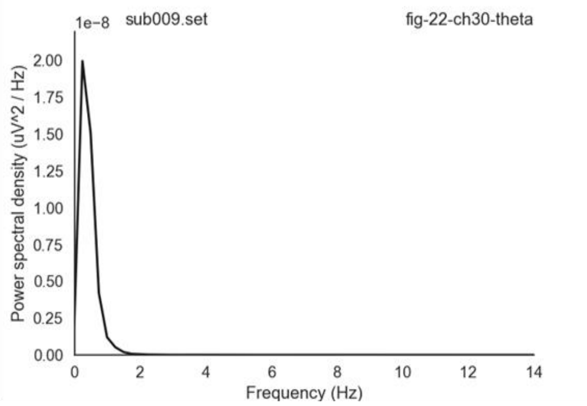
Appendix –C Contains 4 Participants With 26 Attributes for Each

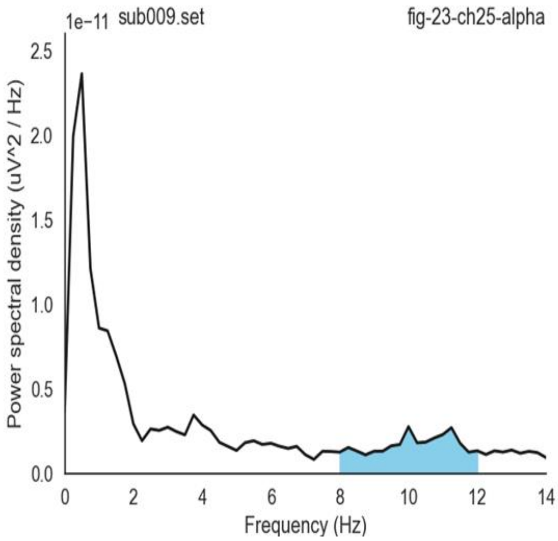
participant number two(sub009.set)

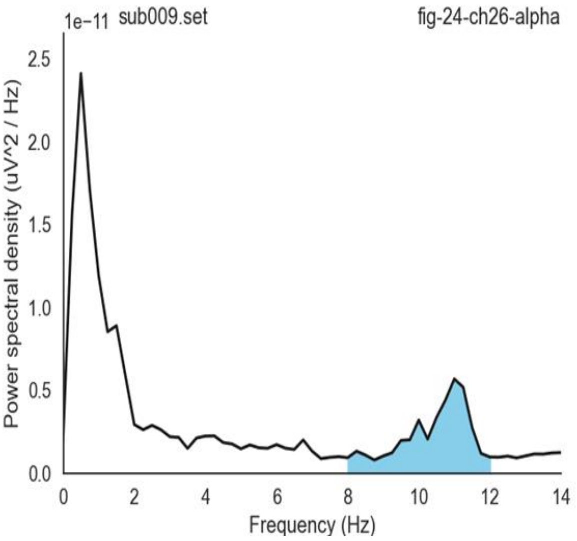




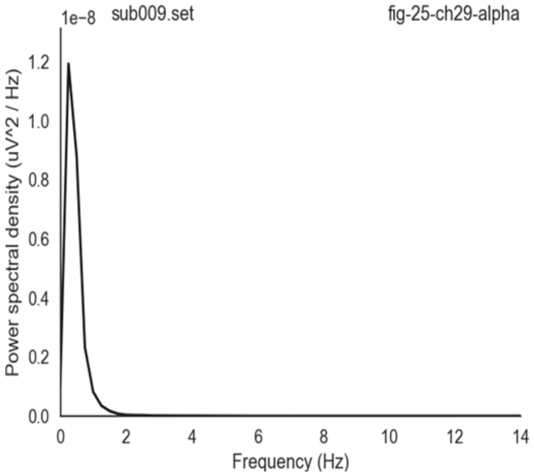


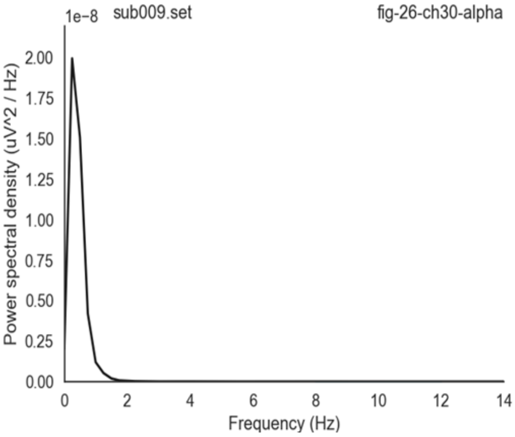






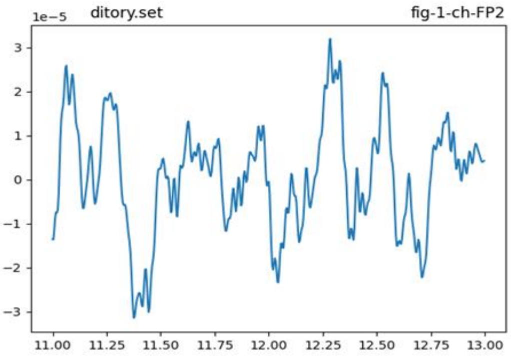
participant Number two (sub009.set)

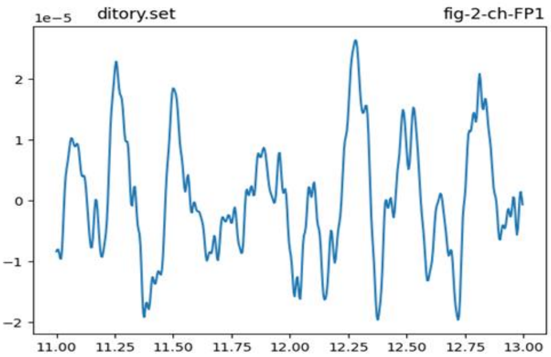


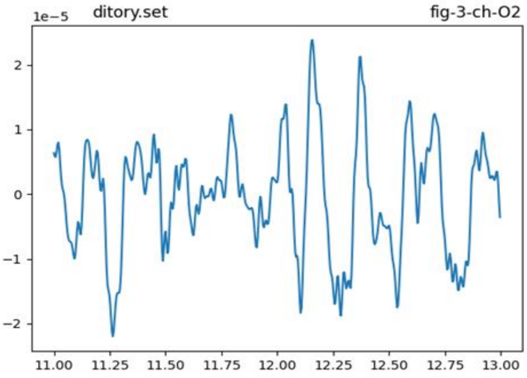


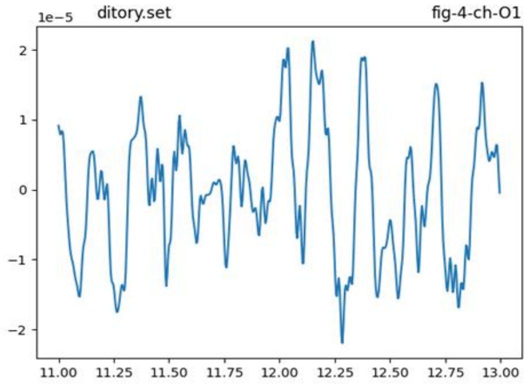
Appendix –C Contains 4 Participants With 26 Attributes for Each

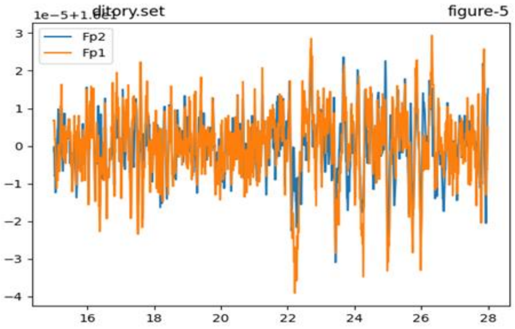
Participant number three (ditory.set)

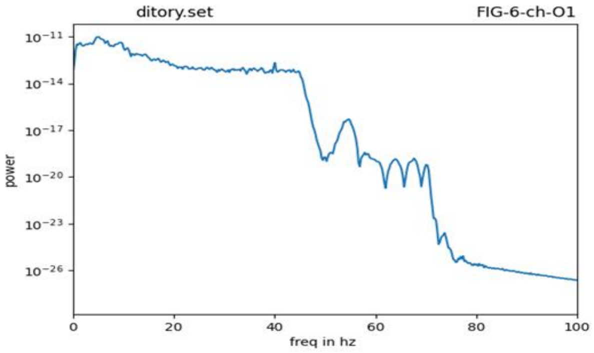






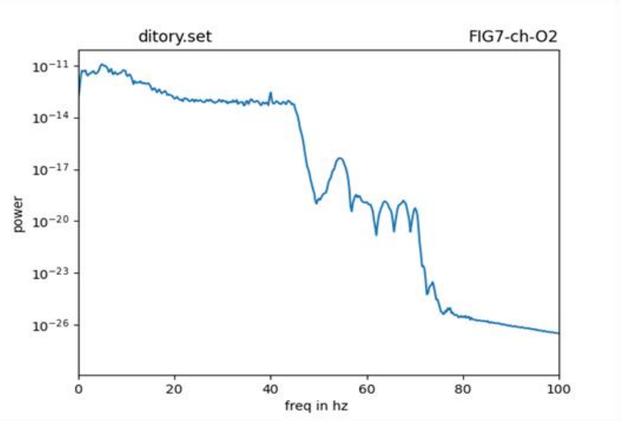


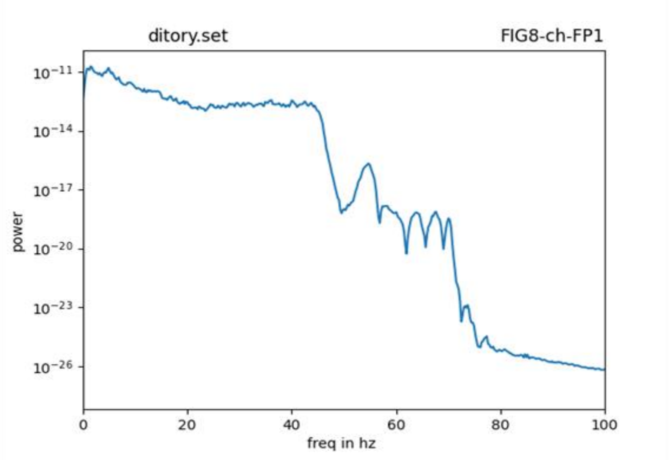


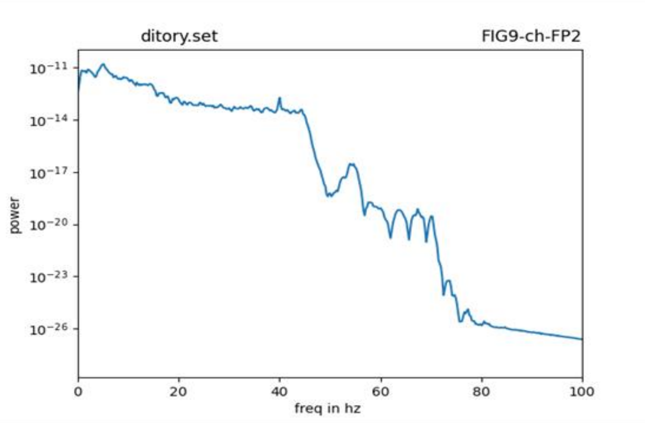


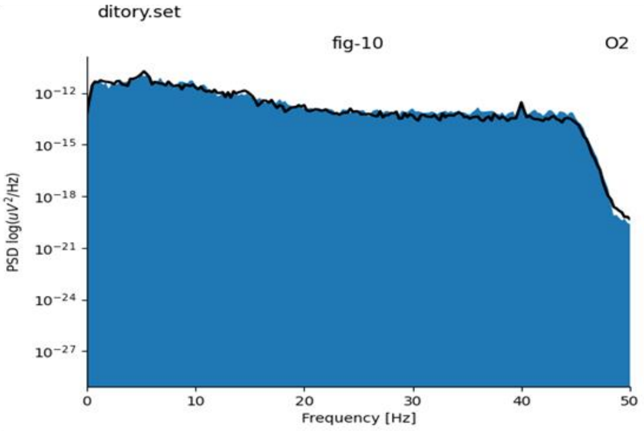
Appendix –C Contains 4 Participants With 26 Attributes for Each

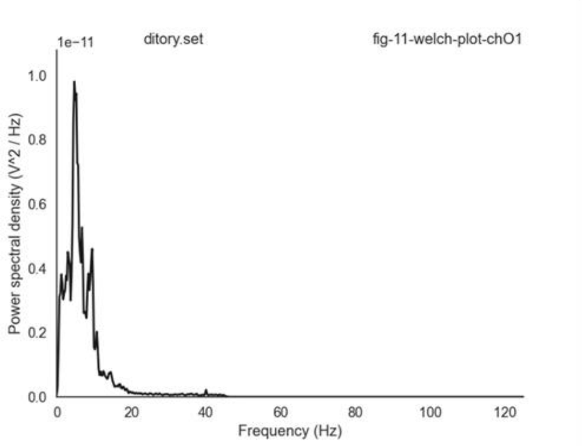
Participant number three (ditory.set

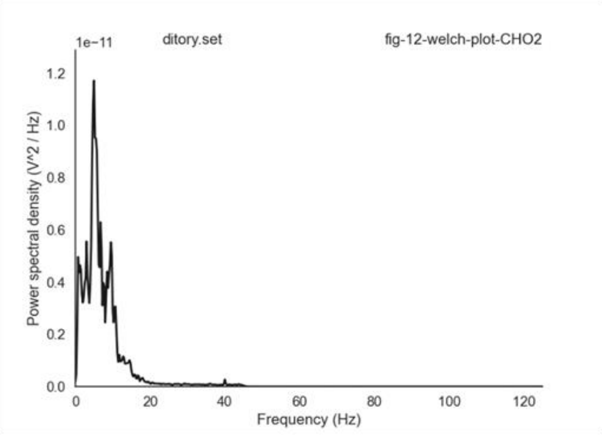






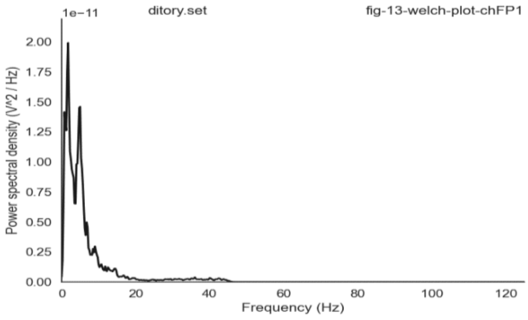


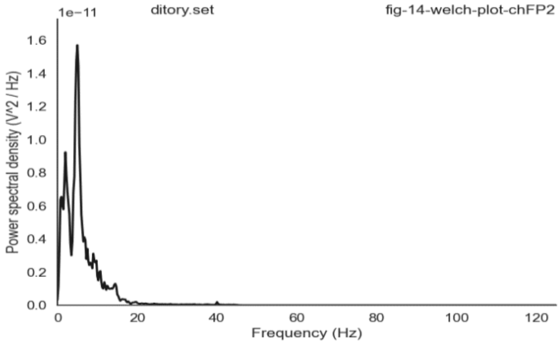


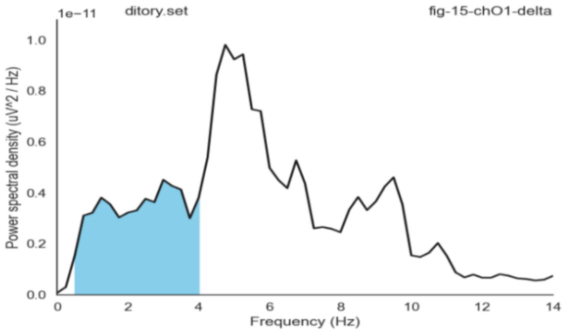


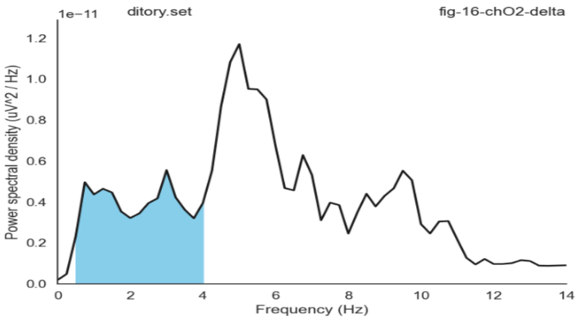
Appendix –C Contains 4 Participants With 26 Attributes for Each

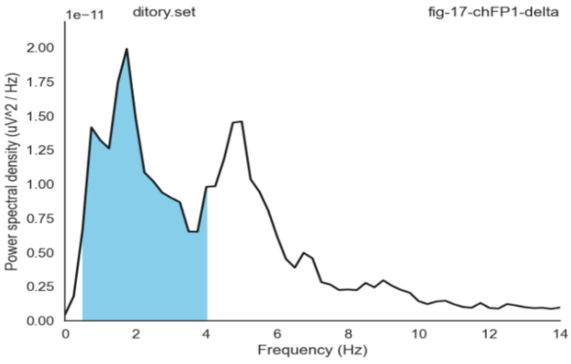
Participant number three (ditory.set)

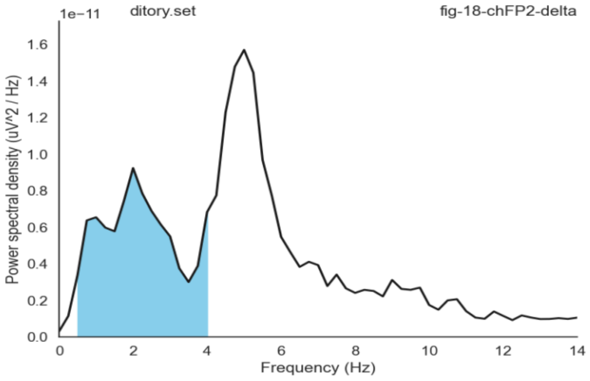






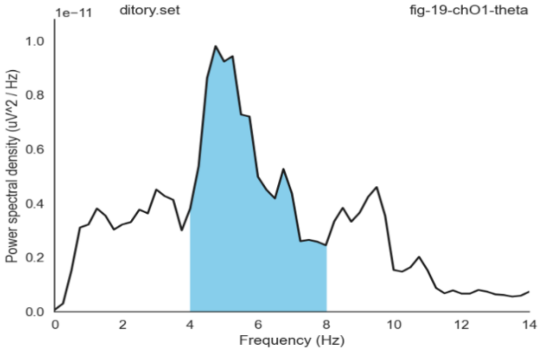


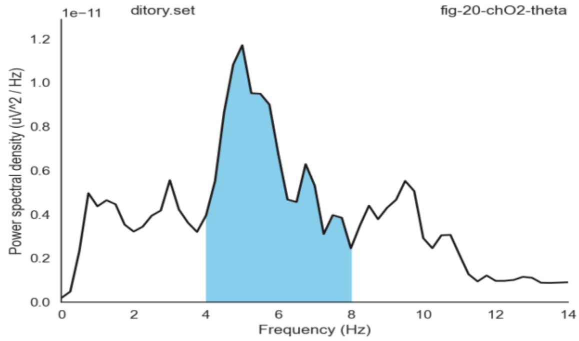


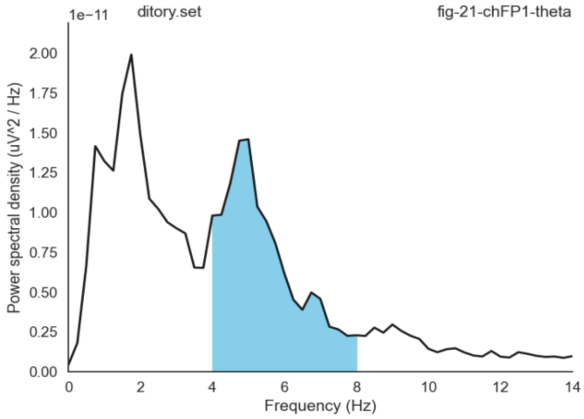


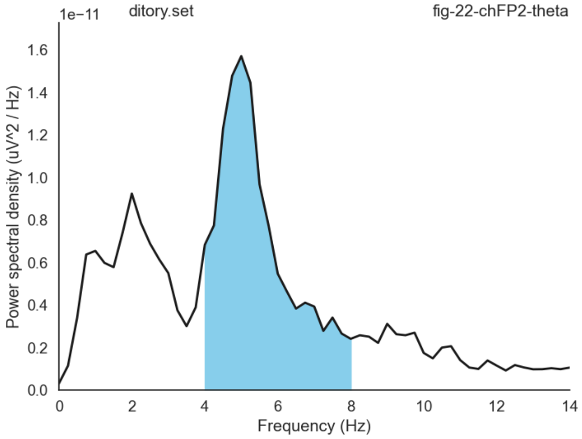
Appendix –C Contains 4 Participants With 26 Attributes for Each

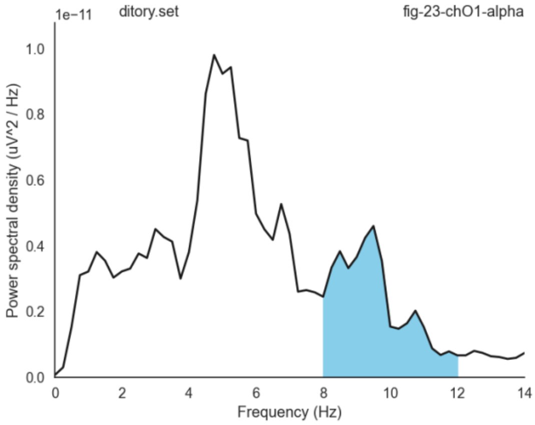
Participant number three (ditory.set)

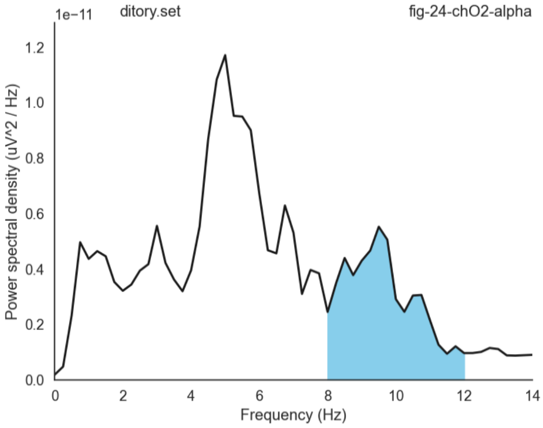






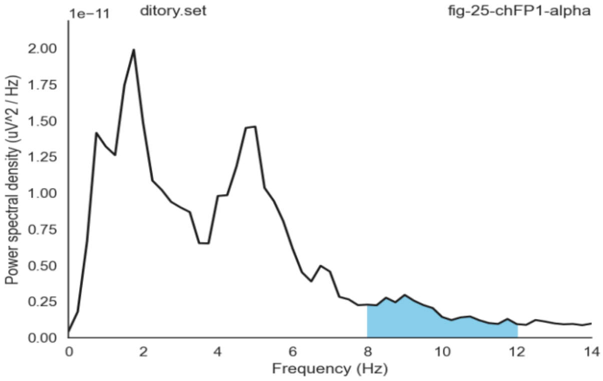


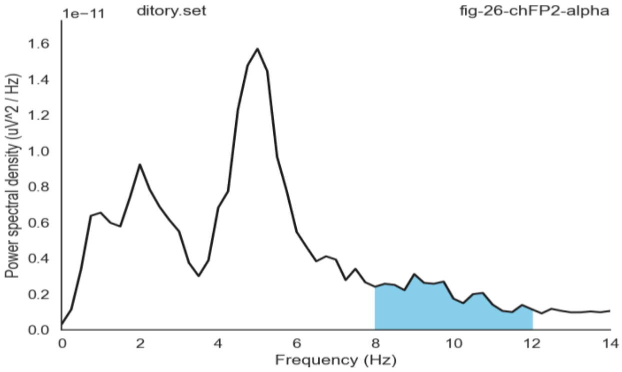




Appendix –C Contains 4 Participants With 26 Attributes for Each

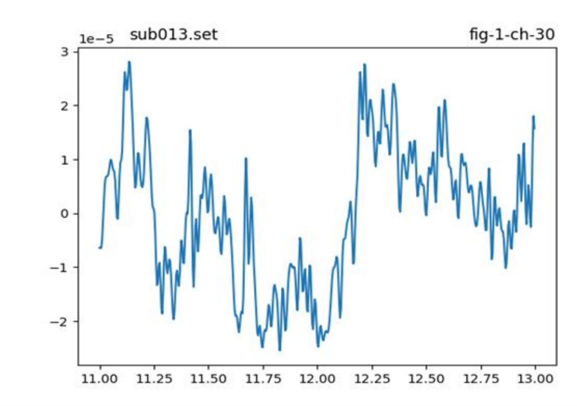
Participant number three (ditory.set)

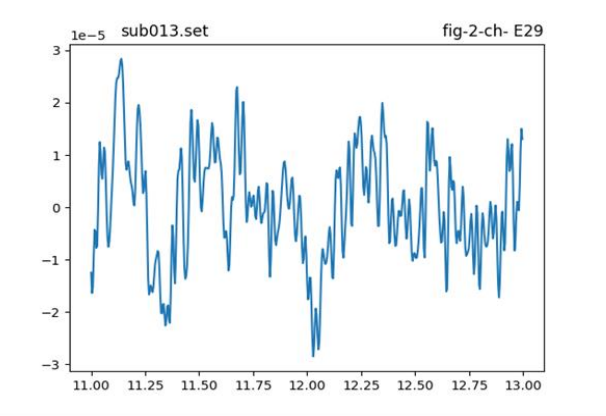


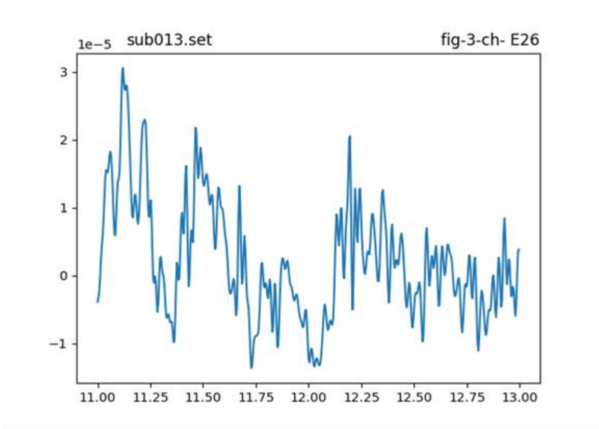


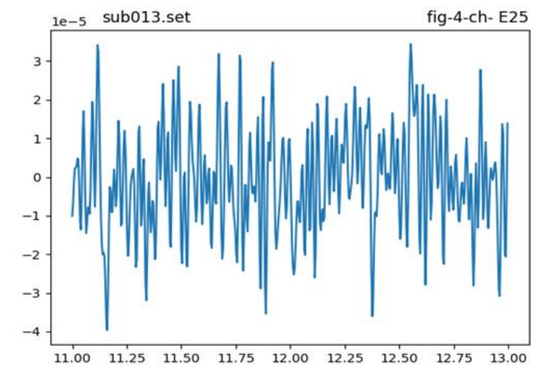
Appendix –C Contains 4 Participants With 26 Attributes for Each

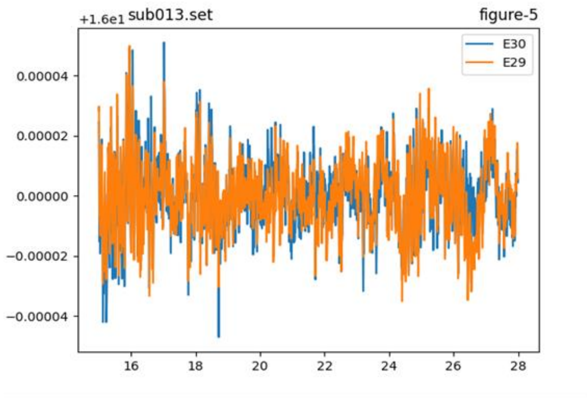
Participant number four (sub013.set), with 26 attributes

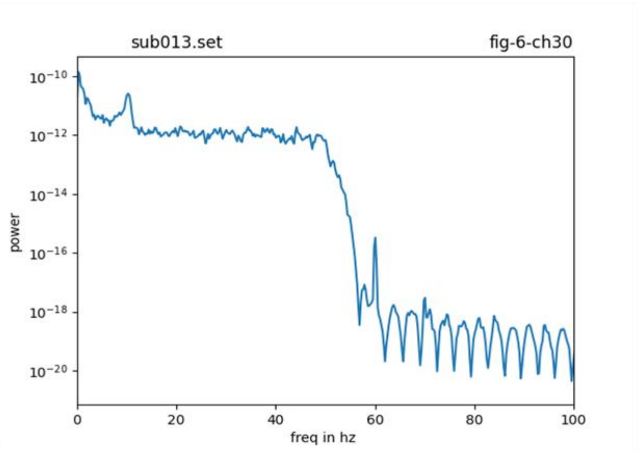






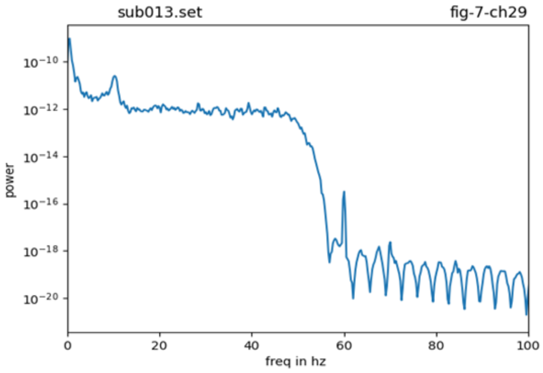


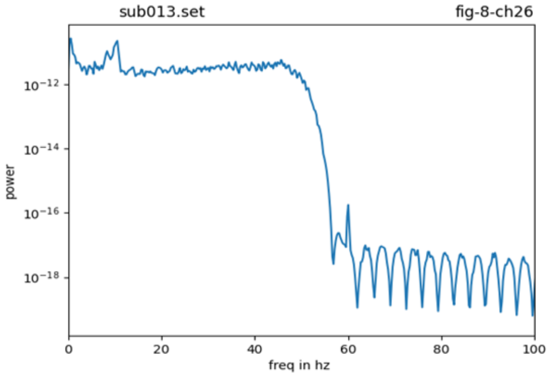


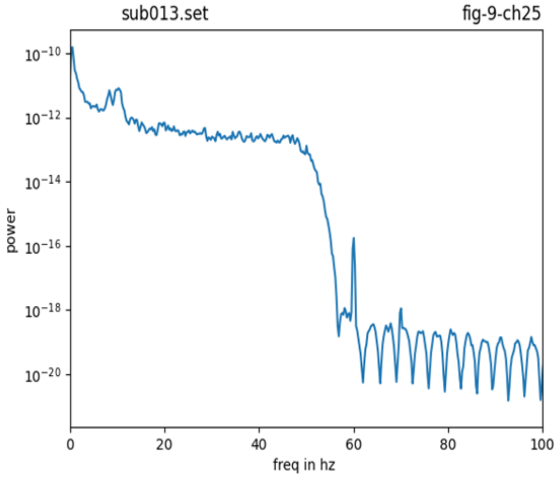


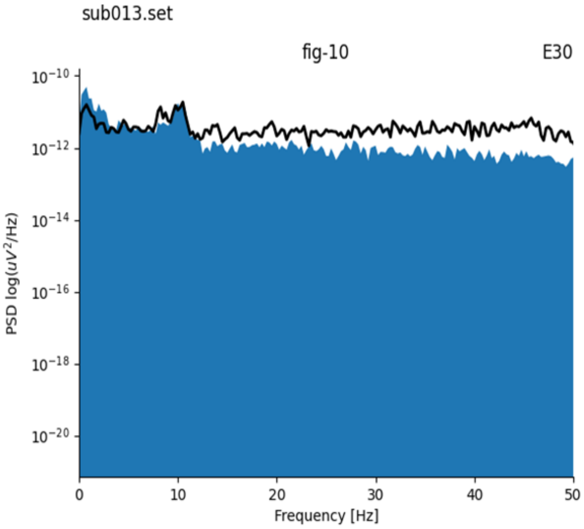
Appendix –C Contains 4 Participants With 26 Attributes for Each

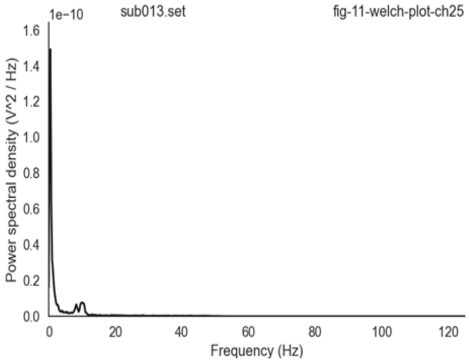
Participant number four (subo13.set)

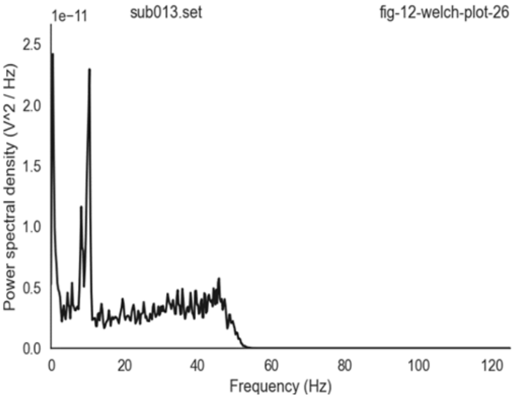






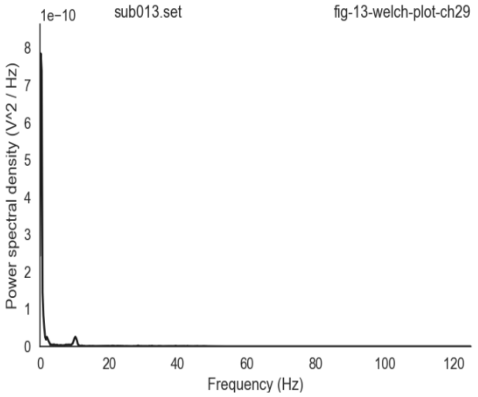


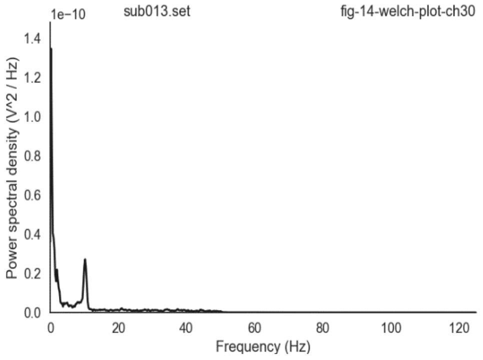


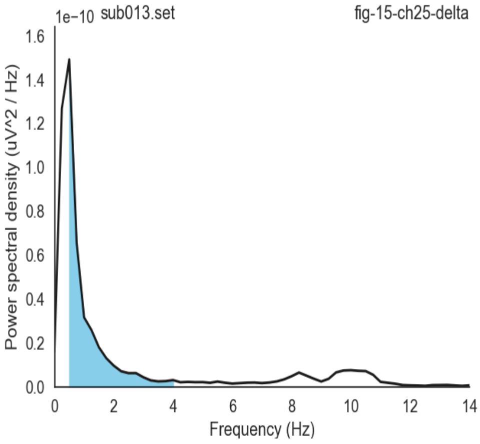


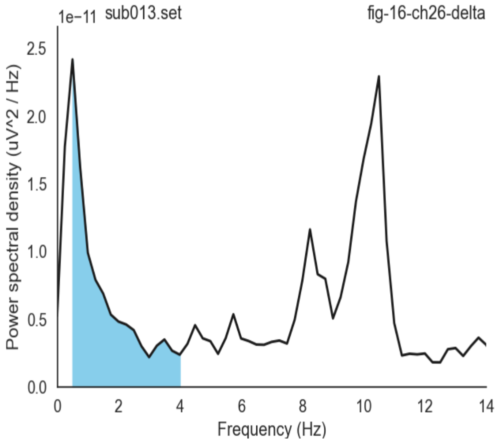
Appendix –C Contains 4 Participants With 26 Attributes for Each

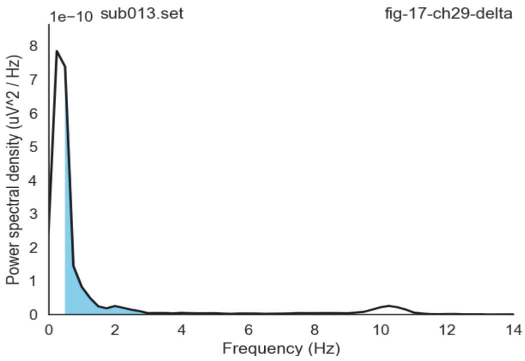
Participant number four (subo13.set)

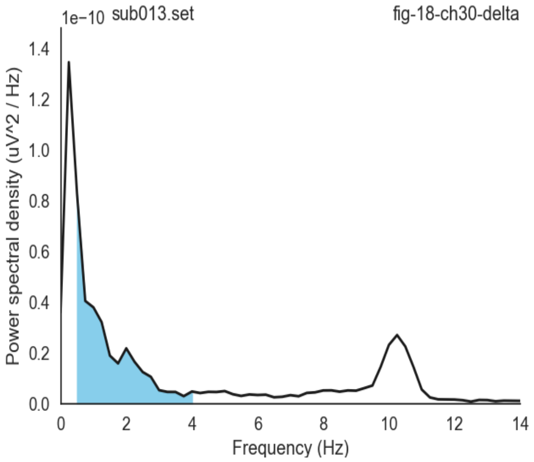






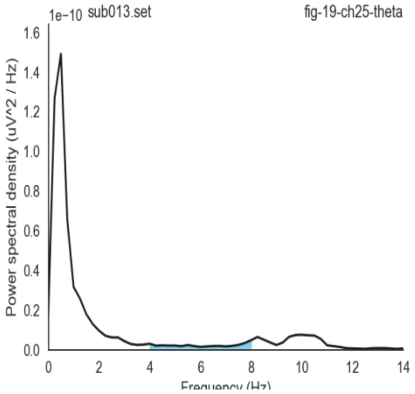


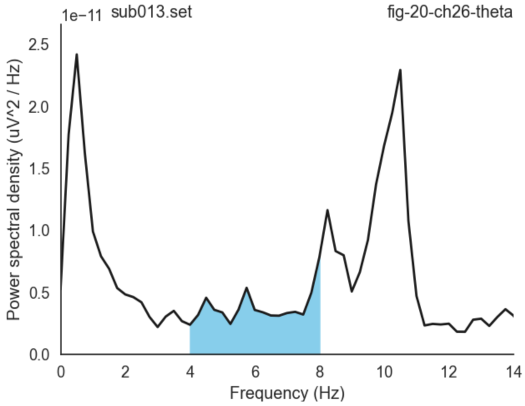


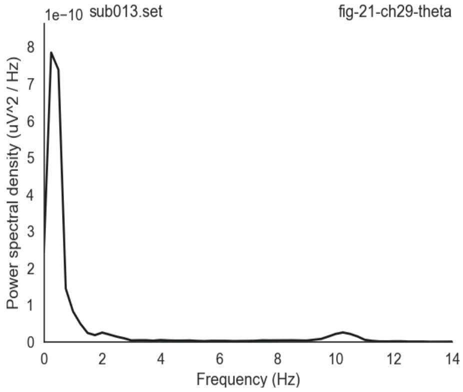


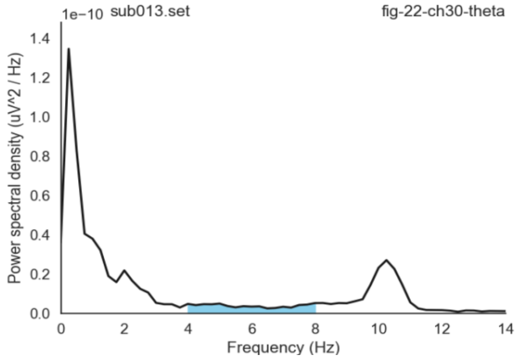
Appendix –C Contains 4 Participants With 26 Attributes for Each

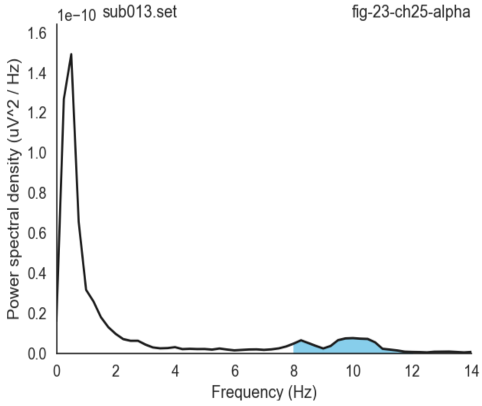
Participant number four (subo13.set)

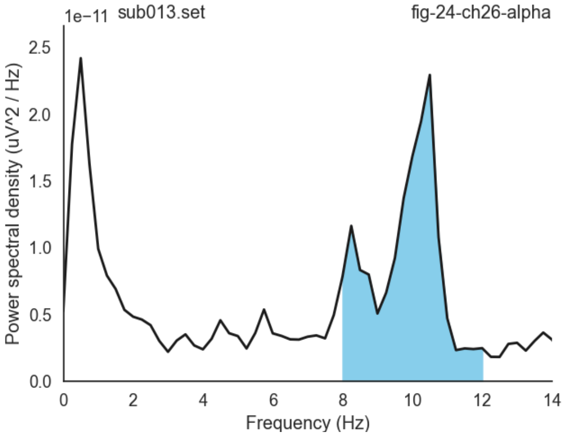




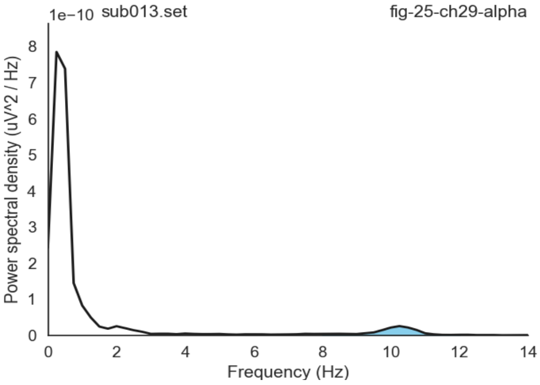


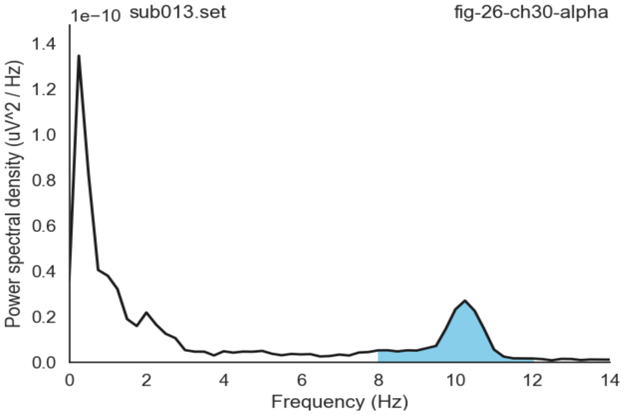






Participant number four (sub013.set)





Appendix (D)

Some Sources Code Program

Written with Python

Appendix-D

1-lin-77.py in that source program refer to linear regression method and show the output in chapter four figure 4.13

```
#linear regression model /lin-77.py

import matplotlib. pyplot as plt

from scipy import stats

x = [5,7,8,7,2,17,2,9,4,11,12,9,6]

y = [99,86,87,88,111,86,103,87,94,78,77,85,86]

slope, intercept, r, p, std_err = stats.linregress(x, y)

def myfunc(x):

 return slope * x + intercept

mymodel = list(map(myfunc, x))

print(‘the data x’,x )

pront(‘the data y’,y)

plt.scatter(x, y)

plt.plot(x, mymodel)

 plt.show()
```

Logistic regression example/log.py

```
import matplotlib.pyplot as plt
import numpy as np
from scipy.special import expit
from sklearn.linear_model import LinearRegression, LogisticRegression
# Generate a toy dataset, it's just a straight line with some Gaussian noise:
xmin, xmax = -5, 5
n_samples = 100
```

```
np.random.seed(0)
```

2- logistic regression program (log.py) source the output refer to chapter four figure 4.14

```
X = np.random.normal(size=n_samples)
y = (X > 0).astype(float)
X[X > 0] *= 4
X += 0.3 * np.random.normal(size=n_samples)
X = X[:, np.newaxis]
# Fit the classifier
clf = LogisticRegression(C=1e5)
clf.fit(X, y)
# and plot the result
plt.figure(1, figsize=(4, 3))
plt.clf()
plt.scatter(X.ravel(), y, label="example data", color="black", zorder=20)
X_test = np.linspace(-5, 10, 300)
loss = expit(X_test * clf.coef_ + clf.intercept_).ravel()
plt.plot(X_test, loss, label="Logistic Regression Model", color="red", linewidth=3)
ols = LinearRegression()
ols.fit(X, y)
plt.plot(
 X_test,
 ols.coef_ * X_test + ols.intercept_,
 label="Linear Regression Model",
 linewidth=1,)
```

Appendix (E)

The source program and the plot results to some image stroke case

Appendix-E

- **More details in that Appendix -E**
- **Explain the yyy-11.py that is referred to in Chapter 3. In that program describe the image and extract the image transform from the original to another form**
- **1-Original, 2-grayscale 3- threshold,4-morphology**
- **In Appendix B explain the source program** *Morplogy-1.py*
- **,in that program explain the morphology feature and how to extract from the original image**
- **In that appendix attachment the main program that extracts the feature (feature-77.py) in the image processing path.**
- **Here collect five images each one containing six steps of attributes that can be illustrated in the following table**

Table E-1 stroke case steps for each image input

x	Original data	Normalized data	Grayscale data	Threshold data	Averaging data	Morphology Data
Image 1	√	√	√	√	√	√
Image 2	√	√	√	√	√	√
Image 3	√	√	√	√	√	√
Image 4	√	√	√	√	√	√
Image 5	√	√	√	√	√	√

- **The list of source program (yyy-11.py) is as follows**

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('F:\image-data/down-6.jpeg')
img=cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
plt.figure(figsize=(8,8))
plt.imshow(img)
plt.axis('off')
plt.title("Original Image")
plt.show()
print('orign image data',img)
print()

# Converting too Grayscale
# To make future image processing less complex and simple we will be converting
# the image loaded in the previous step to grayscale image using the code mentioned below.
# The output image is also displayed below the code.



gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.figure(figsize=(8,8))
plt.imshow(gray,cmap="gray")
plt.axis('off')
plt.title("GrayScale Image")
plt.show()
print('the graycale image ', gray)
print()
print('##########################')



# Converting to a Binary Inverted Image
# To study the image in more detail and have a very precise study of the
# image we will be converting the image into
# a binary inverted image using the code mentioned below.
# The output is also displayed along with the code.
ret, thresh = cv2.threshold(gray, 0, 255,cv2.THRESH_BINARY_INV
+cv2.THRESH_OTSU)
plt.figure(figsize=(8,8))
plt.imshow(thresh,cmap="gray")
plt.axis('off')
plt.title("Threshold Image")
plt.show()
print('the thwrshold image ',thresh)
print()
print('#######################')



# Segmenting the Image
# Now the last step is to get the segmented image with the help of the
# code mentioned below. We will be making use of all the previous
# images somewhere or the other to try to get the most accurate segmented image we can.



kernel = np.ones((3, 3), np.uint16)
closing = cv2.morphologyEx(thresh, cv2.MORPH_CLOSE,kernel, iterations = 15)
bg = cv2.dilate(closing, kernel, iterations = 4)
dist_transform = cv2.distanceTransform(closing, cv2.DIST_L2, 0)
ret, fg = cv2.threshold(dist_transform, 0.02*dist_transform.max(), 255, 0)
cv2.imshow('image', fg)
plt.figure(figsize=(8,8))
plt.imshow(fg)
#plt.imshow(fg,cmap="gray")
plt.axis('off')
plt.title("Segmented Image")
plt.show()
print('the seg. image ', fg)
print('#########################')

plt.figure (figsize=(10, 10))

plt.subplot (2, 2, 1)
plt.axis ('off')
plt.title ("Original Image")
plt.imshow (img, cmap="gray")

plt.subplot (2, 2, 2)
plt.imshow (gray, cmap="gray")
plt.axis ('off')
plt.title ("GrayScale Image")

plt.subplot (2, 2, 3)
plt.imshow (thresh, cmap="gray")
plt.axis ('off')
plt.title ("Threshold Image")

plt.subplot (2, 2, 4)
# plt.imshow (fg, cmap="gray")
plt.imshow (fg)
plt.axis ('off')
plt.title ("Segmented Image")
plt.show ()

# //////////////////////////////////////////////////////////////////////
```

- **<u>The list of the program (morplogy-1.py)</u>**

```
import glob
import os

import cv2
import matplotlib.pyplot as plt
import numpy as np
# mg_dir = ("F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-002/") # Enter Directory of all images
# data_path = os.path.join(mg_dir,'0')
# data_path = os.path.join(mg_dir,'*g')
# files = glob.glob(data_path)
```

```
img=cv2.imread('F:\image-data/down-7.jpeg')
#cv2.imshow('orig',img)
ker=np.ones((3,3),np.uint8)
# erosin=cv2.erode(img,ker,iterations=1 )
# dil=cv2.dilate(img,ker,iterations=1)
plt.Figure(figsize=(10,7))
# plt.subplot(4,2,1)
plt.title('origen image ')
plt.imshow(img)
plt.show()
# # #cv2.imshow('ersion',erosin)
# plt.subplot(4,2,2)
#plt.imshow(erosin)
# # #cv2.imshow('orig',img)
#cv2.waitKey(0)
# plt.title('image ersion ')
# plt.show()



#//////// dilatation /////
#plt.subplot(2,4,3)

#dil=cv2.dilate(img,ker,iterations=1)
# plt.subplot(4,2,3)
# plt.title('image dilation ')
# plt.imshow(dil)
# # plt.subplot(6,2,4)
# # plt.imshow(dil)
# #figure.tight_layout()
# plt.show()
#///////////////////////
#///////// opining //////////
```

```
#plt.Figure(figsize=(8,5))
# plt.subplot(4,2,4)
mor=cv2.morphologyEx(img,cv2.MORPH_OPEN,ker)
plt.imshow(mor)
# plt.subplot(4,2,4)
# plt.imshow(opi)
plt.title('image morpology ')
plt.show()

#ppppppppppppppppppppppppppppppppppppp

img2=cv2.imread('F:\image-data/down-6.jpeg')
plt.imshow(img2)
plt.title('origen img2')
plt.show()
mor1=cv2.morphologyEx(img2,cv2.MORPH_OPEN,ker)
plt.imshow(mor1)
plt.title('morpology img2 ')
plt.show()

#77777777777777777777777777777777777777777
img3=cv2.imread('F:\image-data/ima99.jpeg')
plt.imshow(img3)
plt.title('origen img3')
plt.show()
mor1=cv2.morphologyEx(img3,cv2.MORPH_OPEN,ker)
plt.imshow(mor1)
plt.title('morpology img3 ')
plt.show()
```

Appendix E----------------------The source program and the plot results to some image stroke case

- **The list source python program (feature-77.py), includes feature attributes**

Return to Figure 3.14

```
import cv2
import os
import glob
import matplotlib.pyplot as plt

import numpy as np
mg_dir=('F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004') # /// enter directory of all images
# mg_dir = ("F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-002/") # Enter Directory of all images
# data_path = os.path.join(mg_dir,'0')
data_path = os.path.join(mg_dir,'*g')
files = glob.glob(data_path)

orarray = []# //// original data ////
norarray =[]# //// normalization data /////
grayarray =[]# //// gray scale data ////
thrarray=[]# //// thr. data /////
segarray=[]# //// segm. data /////
avarging=[]# //// avarg. data ////
morarray=[]# //// morp. data
ct=0

for fb in files:
print('the fb',fb)
img = cv2.imread(fb) # get image from the folder image-data
 orarray.append(img)# collect origen data to (orarray )
 # ///// normalization image ////
 img_norm=cv2.normalize(img,0,1,0,cv2.NORM_MINMAX,dtype=cv2.CV_32F)
 norarray.append(img_norm) # collect normalized data in array (norarray)
```

```
# ///// gray scal data ///////
graydata=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
grayarray.append(graydata) # collect gray scale data in array (grayarray)

# ////// thr. data //////

ret,thr=cv2.threshold(graydata,0,255,cv2.THRESH_BINARY_INV+cv2.THRESH_OTSU)
thrarray.append(thr)# collect thr. data in array (thrarray)

#//// seg. data //////////
kernel = np.ones ((3, 3), np.uint8)
closing = cv2.morphologyEx (thr, cv2.MORPH_CLOSE, kernel, iterations=25)
# bg = cv2.dilate (closing, kernel, iterations=4)
dist_transform = cv2.distanceTransform (closing, cv2.DIST_L2, 0)
ret, fg = cv2.threshold (dist_transform, 0.03 * dist_transform.max (), 255, 0)
segarray.append(fg)

#//// avar. data /////////

img2 = cv2.cvtColor (img, cv2.COLOR_BGR2RGB) # Fixes color read issue
av3 = cv2.blur (img2, (3, 3))
avarging.append(av3)

# //// morphology data /////

morp = cv2.morphologyEx (img, cv2.MORPH_OPEN, kernel)
morarray.append(morp)



if ct==10:
# if fb == 'F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-002\img-00002-00005.jpg': # the counter limit
```

```
break
print ('the orignal data',orarray)# display the origen data of image
print()
print ('the norm data', norarray)# display that normalize data
print()
print('the garayscal data',grayarray)# display gray scale data
print()
print('the thr data ',thrarray)# display the threshold data
print()
print ('the seg. image ', segarray) # display the seg. data
print()
print('the avar. data ',avarging)
print()
print('the morphology data ',morarray)

# //// display . plot the image /////
plt.imshow (img) # display the original image
plt.suptitle(fb)
plt.title('original image ')
# plt.text(230,25,'orign data')
plt.show ()

# //// display the normalized image////
plt.imshow (img_norm)
plt.suptitle(fb)
plt.title('normalized image')
# plt.text(230,25,'norm')
plt.show ()

# ///// display the gray scal image //////
plt.imshow (graydata) # display the gray image
plt.suptitle(fb)
plt.title('grayscale image ')
```

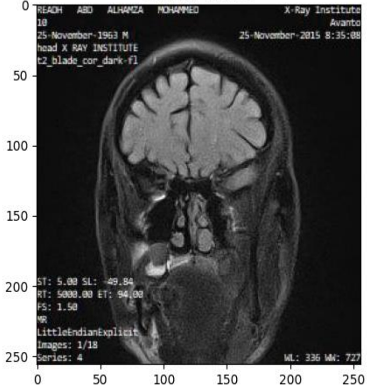
```
 # plt.text(230,25,'gray')
 plt.show ()
 # //// display the thr. image ////

 plt.imshow (thr) # display the thr image
 plt.suptitle (fb)
 plt.title ('thr. image ')
 # plt.text(230,25,'gray')
 plt.show ()

 plt.imshow (fg) # display the thr image
 plt.suptitle (fb)
 plt.title ('seg. image ')
 # plt.text(230,25,'gray')
 plt.show ()

# ///// Averaging image display ///////
 plt.imshow (av3)
 plt.suptitle (fb)
 plt.title('Averaging - 3x3')
 ct = ct + 1
 print ('the count', ct)
 plt.show()

 # //////morphology image display /////

 # img2=cv2.imread('F:\image-data/down-6.jpeg')
 plt.imshow(morp)
 plt.suptitle(fb)
 plt.title('morphology image ')
 plt.show()
print('exit')
print(ct)
```

Table E-2 image processing steps

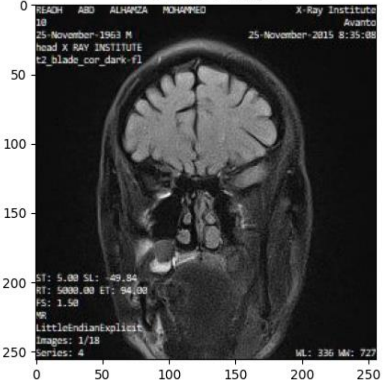
	Imege-1	Image-2	Image-3	Image-4	Image-5
Cas1-1	Orig.	Orig.	Orig.	Orig.	Orig.
Case-2	Norm.	Norm.	Norm.	Norm.	Norm.
Cass-3	Gray.	Gray.	Gray.	Gray.	Gray.
Case-4	Thresh.	Thresh.	Thresh.	Thresh.	Thresh.
Case-5	Aver.	Aver.	Aver.	Aver.	Aver.
Case-6	Morph.	Morph.	Morph.	Morph.	Morph.

The above table shows the total image is 5, and 30 items of the total, so this is a small sample of that work that contains 1000 samples (image), Figure E-1 display all images

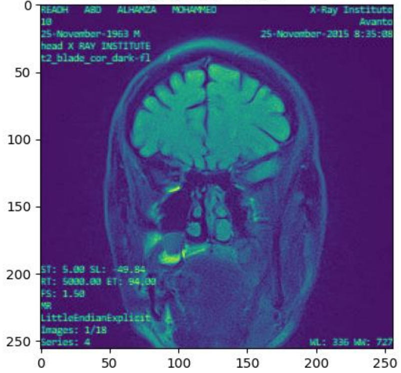
F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00001.jpg
original image



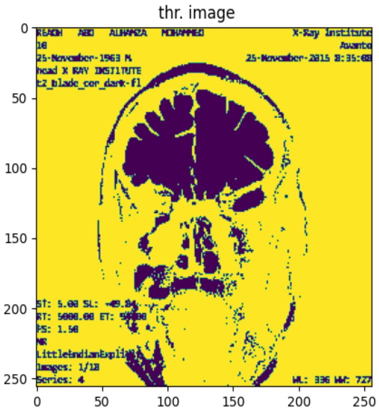
F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00001.jpg
normalized image



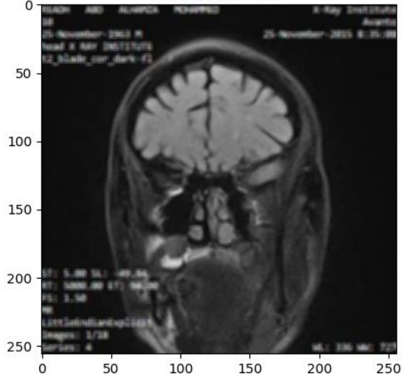
F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00001.jpg
grayscale image



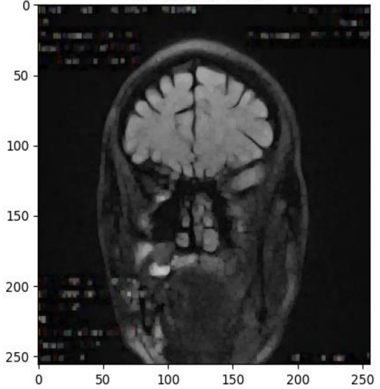
F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00001.jpg
thr. image

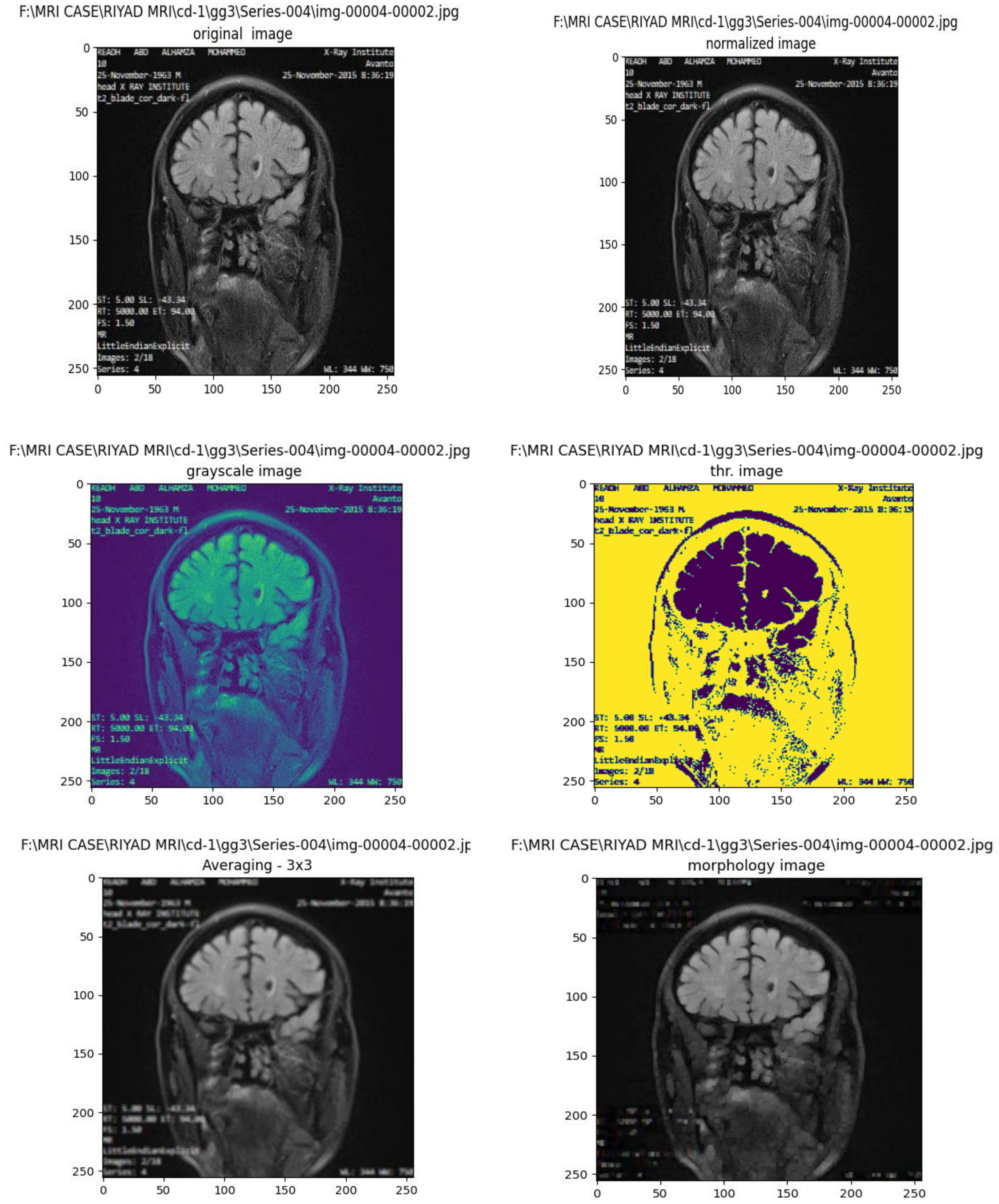


F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00001.jpg
Averaging - 3x3



F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00001.jpg
morphology image





F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00002.jpg
original image

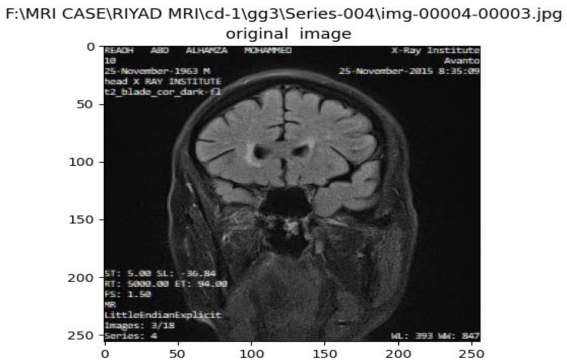
F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00002.jpg
normalized image

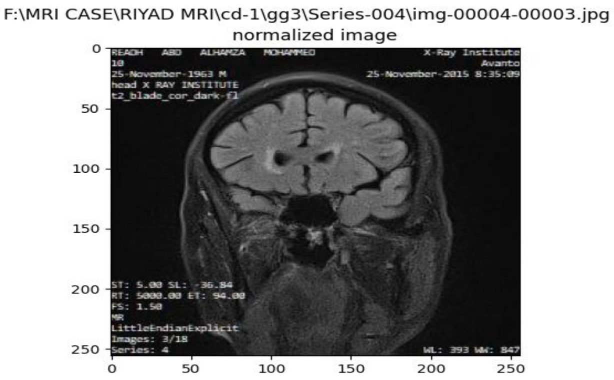
F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00002.jpg
grayscale image

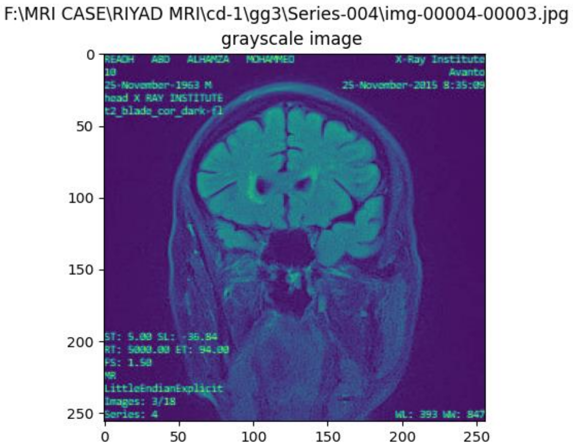
F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00002.jpg
thr. image

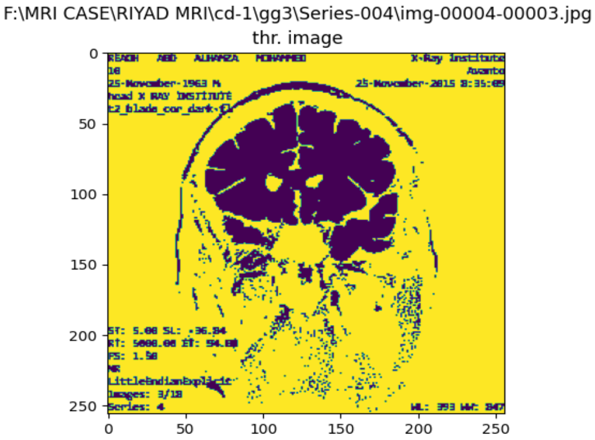
F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00002.jp
Averaging - 3x3

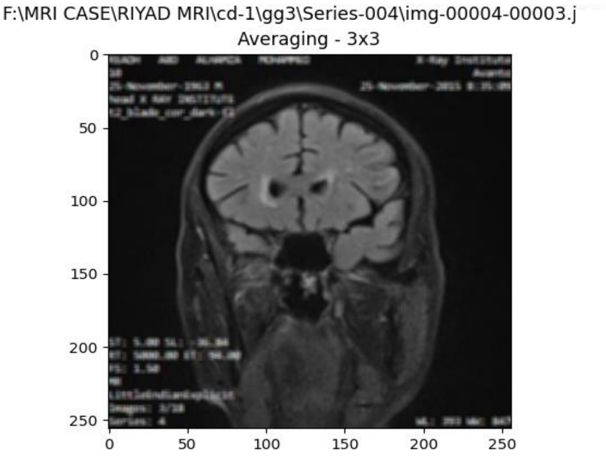
F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00002.jpg
morphology image

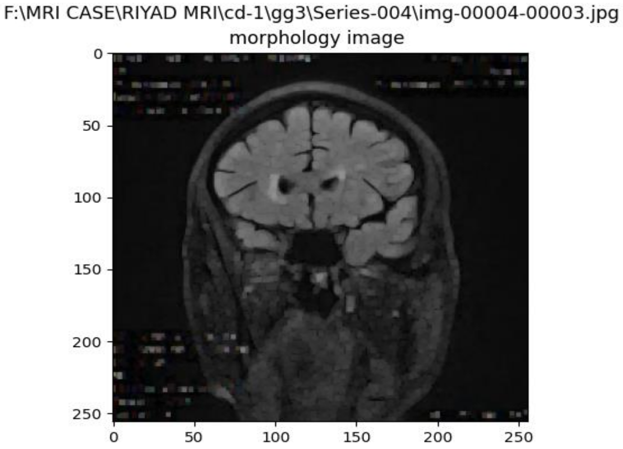

F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00003.jpg
original image

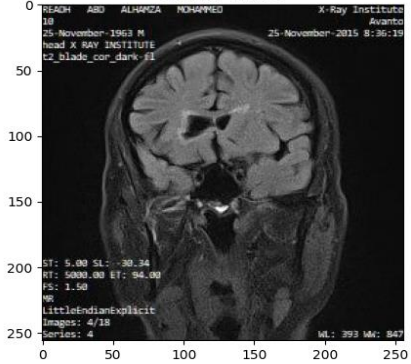

F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00003.jpg
normalized image

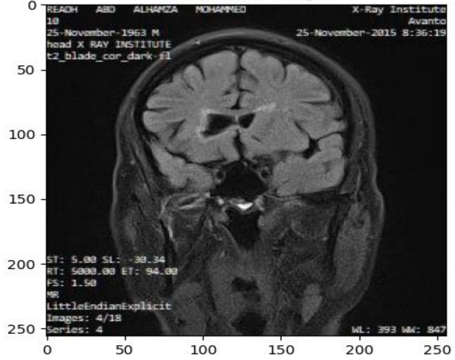

F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00003.jpg
grayscale image

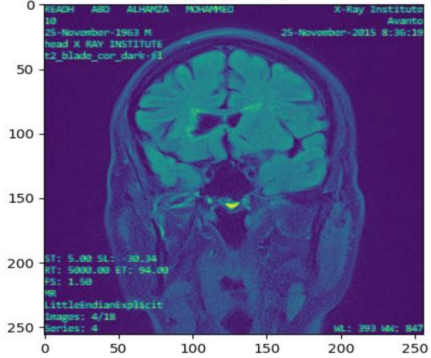

F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00003.jpg
thr. image

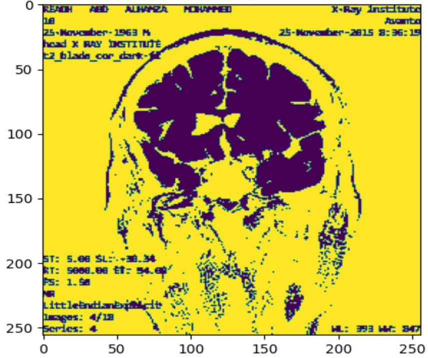

F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00003.j
Averaging - 3x3

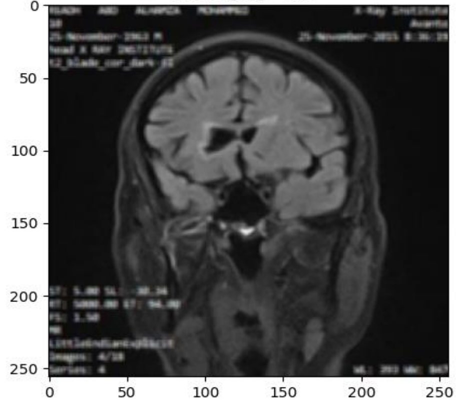

F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00003.jpg
morphology image

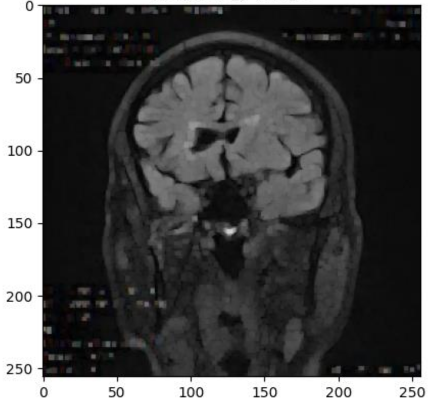

F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00004.jpg
original image


F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00004.jpg
normalized image


F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00004.jpg
grayscale image


F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00004.jpg
thr. image


F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00004.jpg
Averaging - 3x3


F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00004.jpg
morphology image

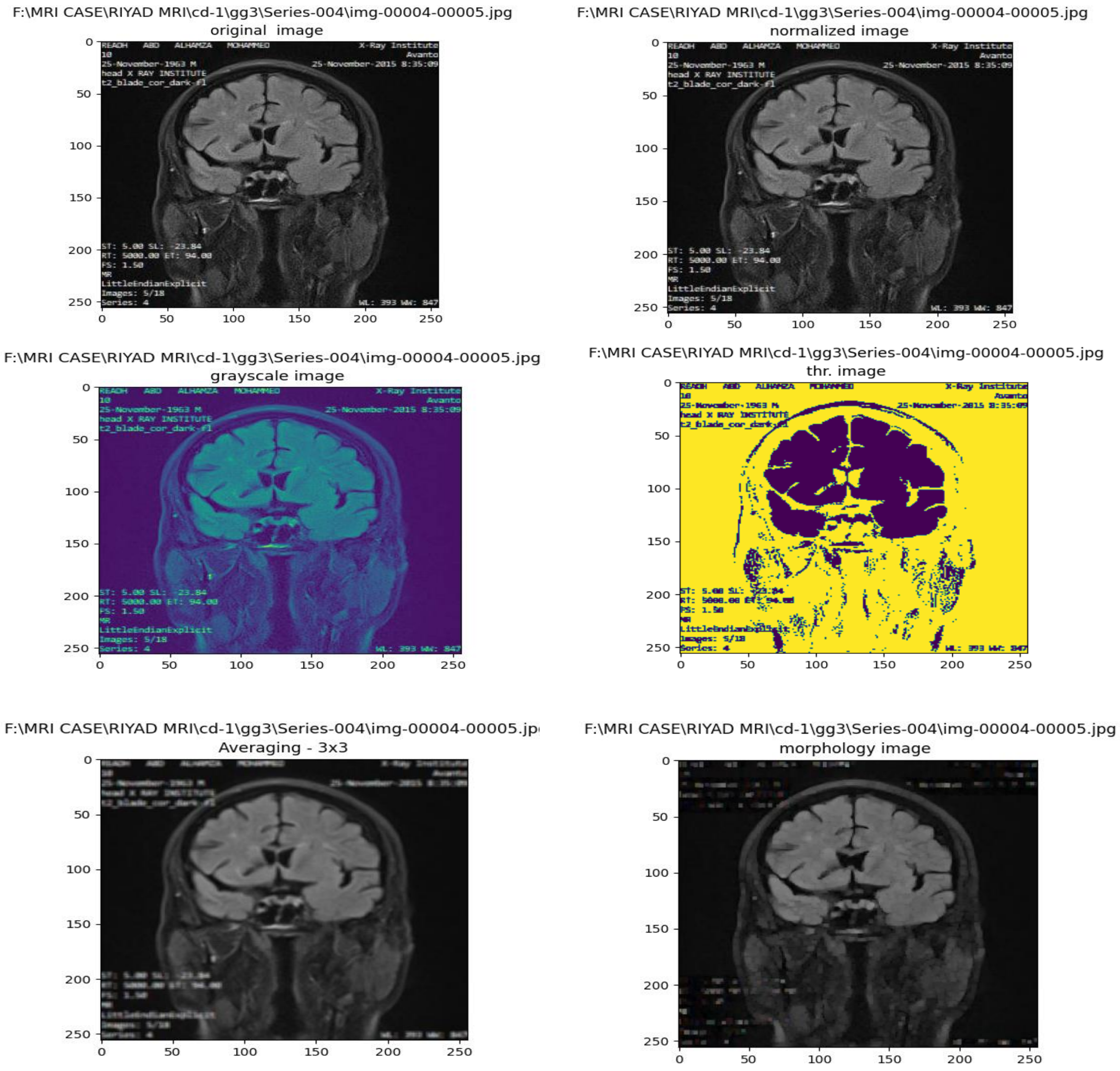

F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00005.jpg
original image
F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00005.jpg
normalized image
F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00005.jpg
grayscale image
F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00005.jpg
thr. image
F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00005.jp
Averaging - 3x3
F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00005.jpg
morphology image
0
50
100
150
200
250

Figure E-1 shows the image processing steps

In that Figures show the plots of six steps attributes of image

1. Original image
2. Normalize image
3. Grayscale image
4. Threshold image
5. Averaging image
6. Morphology image

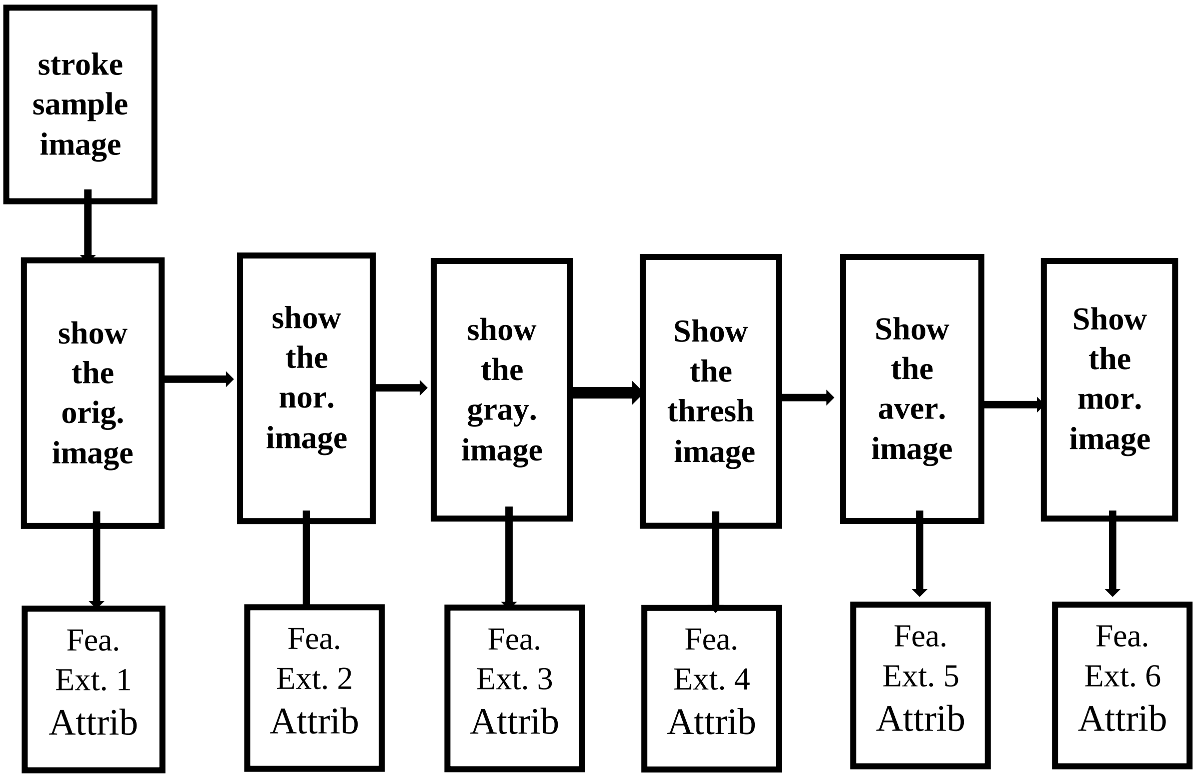


Figure E-2 The Processing Sequence Steps Flow of any Image with features extraction

Note:

Fea---- refer to features

Ext. ---- refers to the extraction

Attrib. ---- refer to attributes

Orig.------ refer to original

Thresh ---- refer to threshold

Mor. ---- refer to morphology

Appendix (F)

Image Data with The Source Program

Appendix-F

- Some Details About the Appendix -F
    - ✓ Here list the source Python code imdata.py, with flowchart, figure-E-1
    - ✓ Contain ten image each one including size w=9, h=9 ,ch=3
    - ✓ The total of element 243 for each image
    - ✓ Each image represents the view of the MRI scan for a high-resolution device.
    - ✓ Each image refers to the original of the brain stroke.
- The files name: the fb F:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004\img-00004-00001.jpg,00004-0002,00004-00003,00004-00004,00004-00005,00004-00006,00004-00005,00004-00007,00004-00008,00004-00009
- The original size of the image (256, 256, 3) for each original image
- the count 0 --- flag counter of image
- the resize image (9, 9, 3)
- channel=3, w=9, h=9
- hear print the data **for 3 images** only after resizing the original image into 9x9x3 which has 243 elements for each image.
- **THE table F-1** INCLUDE 3 PART TABL, A, B,C each table contain 243 value representant the image value ***after resize***
- And the overall image value 720 value
- Table F-1 (A, B, C) represent image pixel value after resize.

  The list program imdata.py source code

```
import cv2
import os
import glob
import matplotlib.pyplot as plt

import numpy as np
mg_dir=('G:\MRI CASE\RIYAD MRI\cd-1\gg3\Series-004')
data_path = os.path.join(mg_dir,'*g')
files = glob.glob(data_path)

orarray = []     # //// original data ////
norarray =[]     # //// normalization data /////
grayarray =[]     # //// gray scale data ////
thrarray=[]     # //// thr. data /////
segarray=[]     # //// segm. data /////
avarging=[]     # //// avarg. data ////
```

```
morarray=[]     # //// morp. data
ct=0

for fb in files:

     print('the fb',fb)
     img = cv2.imread(fb)   # get  image from the  folder
     print('the size ',img.shape)
if ct==10:

        break
     ff3=cv2.resize(img,(9,9))
     print('the coun',ct)
     print('the resize image',ff3.shape )
     print('the values\n',ff3)
     plt.imshow (img )  # display the original image
     plt.suptitle(fb)
     plt.title('original  image ')
     plt.show ()
     ct=ct+1



print('exit')
print(ct)
```

Table F-1-A	channel 1			Channel 2			Channel 3		
IMA-1	CH-1-1	CH1-2	CH1-3	CH2-1	CH2-2	CH2-3	CH-3=1	CH3-2	CH3-3
VAL-1	10	12	15	13	14	15	14	14	15
VAL-2	11	13	13	12	13	16	13	13	17
VAL-3	13	12	13	13	13	16	112	114	112
VAL-4	164	163	158	35	35	35	14	14	14
VAL-5	158	159	164	37	37	37	12	12	12
VAL-6	30	30	30	13	13	13	14	14	14
VAL-7	15	15	15	25	25	25	28	28	28
VAL-8	14	14	14	22	22	22	17	17	17
VAL-9	66	66	66	112	112	112	14	14	14
VAL-10	14	14	14	20	20	20	18	18	15
VAL-11	17	17	17	143	143	143	14	14	14
VAL-12	27	27	27	57	57	57	64	64	64
VAL-13	13	13	13	135	135	135	14	14	14
VAL-14	16	16	16	163	163	163	13	13	13
VAL-15	25	25	25	52	52	52	16	16	16
VAL-16	14	14	14	129	129	129	47	47	47
VAL-17	17	17	17	127	127	127	16	16	16
VAL-18	138	138	138	127	127	127	12	12	12
VAL-19	14	14	14	37	37	37	52	52	52
VAL-20	16	16	16	103	103	103	16	16	16
VAL-21	44	44	44	42	42	42	14	14	14
VAL-22	11	10	10	74	74	74	41	41	41
VAL-23	22	20	20	45	45	45	15	15	15
VAL-24	73	73	73	44	44	44	15	15	15
VAL-25	149	158	158	57	57	57	45	45	45
VAL-26	22	23	23	46	46	46	21	15	14
VAL-27	22	20	21	64	64	64	16	14	13

Table F-1-B	**channel 1**			**Channel 2**			**Channel 3**		
IMA-2	CH-1-1	CH1-2	CH1-3	CH2-1	CH2-2	CH2-3	CH-3=1	CH3-2	CH3-3
VAL-1	**13**	**15**	**18**	**13**	**17**	**15**	**13**	**13**	**13**
VAL-2	**13**	**15**	**15**	**18**	**16**	**17**	**9**	**7**	**13**
VAL-3	**16**	**17**	**14**	**14**	**14**	**14**	**113**	**117**	**115**
VAL-4	**167**	**166**	**162**	**36**	**36**	**36**	**22**	**22**	**22**
VAL-5	**151**	**152**	**156**	**105**	**105**	**105**	**15**	**15**	**15**
VAL-6	**16**	**23**	**23**	**23**	**23**	**23**	**13**	**13**	**13**
VAL-7	**12**	**12**	**12**	**154**	**154**	**154**	**19**	**19**	**19**
VAL-8	**16**	**16**	**16**	**150**	**150**	**150**	**16**	**16**	**16**
VAL-9	**31**	**31**	**31**	**151**	**151**	**151**	**13**	**13**	**13**
VAL-10	**17**	**17**	**17**	**157**	**157**	**157**	**66**	**66**	**66**
VAL-11	**20**	**20**	**20**	**135**	**135**	**135**	**15**	**15**	**15**
VAL-12	**114**	**114**	**114**	**120**	**120**	**120**	**13**	**13**	**13**
VAL-13	**16**	**16**	**16**	**16**	**16**	**16**	**72**	**72**	**72**
VAL-14	**13**	**13**	**13**	**13**	**13**	**13**	**19**	**19**	**19**
VAL-15	**68**	**68**	**68**	**68**	**68**	**68**	**15**	**15**	**15**
VAL-16	**16**	**16**	**16**	**49**	**49**	**49**	**52**	**52**	**52**
VAL-17	**18**	**18**	**18**	**128**	**128**	**128**	**16**	**16**	**16**
VAL-18	**16**	**16**	**16**	**106**	**106**	**106**	**14**	**14**	**14**
VAL-19	**149**	**158**	**157**	**52**	**52**	**52**	**144**	**145**	**147**
VAL-20	**154**	**166**	**170**	**55**	**55**	**55**	**154**	**162**	**170**
VAL-21	**130**	**130**	**130**	**71**	**71**	**71**	**130**	**130**	**130**
VAL-22	**106**	**106**	**106**	**16**	**15**	**15**	**91**	**91**	**91**
VAL-23	**15**	**16**	**14**	**10**	**10**	**10**	**44**	**44**	**44**
VAL-24	**13**	**17**	**16**	**52**	**52**	**52**	**71**	**71**	**71**
VAL-25	112	112	113	**68**	**68**	**68**	**19**	**19**	**19**
VAL-26	**13**	**17**	**16**	22	20	21	**106**	**106**	**106**
VAL-27	74	74	73	54	53	52	44	43	44

Table F-1-C	**channel 1**			**Channel 2**			**Channel 3**		
IMA-3	CH-1-1	CH1-2	CH1-3	CH2-1	CH2-2	CH2-3	CH-3=1	CH3-2	CH3-3
VAL-1	**10**	**12**	**15**	**13**	**14**	**15**	**11**	**11**	**11**
VAL-2	**11**	**13**	**13**	**12**	**13**	**16**	**13**	**13**	**17**
VAL-3	**13**	**12**	**13**	**13**	**13**	**16**	**112**	**114**	**112**
VAL-4	**164**	**163**	**158**	**35**	**35**	**35**	**17**	**17**	**17**
VAL-5	**158**	**159**	**164**	**37**	**37**	**37**	**12**	**12**	**12**
VAL-6	**30**	**30**	**30**	**13**	**13**	**13**	**14**	**14**	**14**
VAL-7	**15**	**15**	**15**	**25**	**25**	**25**	**28**	**28**	**28**
VAL-8	**14**	**14**	**14**	**22**	**22**	**22**	**16**	**16**	**15**
VAL-9	**66**	**66**	**66**	**112**	**112**	**112**	**14**	**14**	**14**
VAL-10	**14**	**14**	**14**	**20**	**20**	**20**	**18**	**18**	**15**
VAL-11	**17**	**17**	**17**	**143**	**143**	**143**	**14**	**14**	**14**
VAL-12	**27**	**27**	**27**	**57**	**57**	**57**	**64**	**64**	**64**
VAL-13	**13**	**13**	**13**	**135**	**135**	**135**	**14**	**14**	**14**
VAL-14	**16**	**16**	**16**	**163**	**163**	**163**	18	**18**	**18**
VAL-15	**25**	**25**	**25**	**52**	**52**	**52**	**16**	**16**	**16**
VAL-16	**14**	**14**	**14**	**129**	**129**	**129**	**47**	**47**	**47**
VAL-17	**17**	**17**	**17**	**127**	**127**	**127**	20	20	20
VAL-18	**138**	**138**	**138**	**127**	**127**	**127**	**12**	**12**	**12**
VAL-19	**14**	**14**	**14**	**37**	**37**	**37**	**52**	**52**	**52**
VAL-20	**16**	**16**	**16**	**103**	**103**	**103**	**16**	**16**	**16**
VAL-21	**44**	**44**	**44**	**42**	**42**	**42**	**14**	**14**	**14**
VAL-22	**11**	**10**	**10**	**74**	**74**	**74**	**41**	**41**	**41**
VAL-23	**22**	**20**	**20**	**45**	**45**	**45**	**15**	**15**	**15**
VAL-24	**73**	**73**	**73**	**44**	**44**	**44**	**15**	**15**	**15**
VAL-25	**149**	**158**	**158**	**57**	**57**	**57**	**45**	**45**	**45**
VAL-26	**22**	23	23	**46**	**46**	**46**	**21**	**15**	**14**
VAL-27	95	95	98	113	113	112	120	120	122

الخلاصة

على نطاق العالم ، تُسبِّب الحوادث الدماغية او السكتات الدماغية الغالبية العظمى من الوفيات . ولذلك، فإن أحد مجالات التركيز الرئيسية للدراسة الحالية هو تحديد الأعراض التي قد تحدث قبل حدوث السكتة الدماغية. ويهدف هذا البحث إلى تحسين فعالية العلاجات المنقذة للحياة للأفراد المعرضين لخطر شديد من خلال معالجة المشكلة المذكورة. والهدف من ذلك هو مساعدة الناس على الشعور بالتحسّن والأمل مرة أخرى.

السكتة الدماغية هي واحدة من تلك الأشياء التي يختبرها الكثير من الناس ولا تحضّرك لها حقاً. لذلك، ليس لدى الناس الكثير من الوقت لفعل أي شيء قبل أن يحدث، بما في ذلك التحضير له. وقد يواجه الناجون من الجرأة تغييرات عميقة في مهاراتهم البدنية وتصوراتهم وقدراتهم الإدراكية. قد تحدث امور كثيرة مختلفة للناس بعد السكتة الدماغية ، لكن بعض العواقب تبدو اكثر شيوعا من غيرها . وقد أعد هذا الكتيب بهدف إعلام الأفراد عن السكتات الدماغية وكيفية تأثيرها على حياتهم اليومية.

وتعرف الهجمات على الدماغ طبياً بالسكتات الدماغية. نقص الأكسجين يصل إلى أنسجة الدماغ مسبباً هذه الحالة مـر مـن وقد يكون الشخص مصاباً بسكتات دماغية ونزفية على حد سواء، وهما نوعان فرعيان من نفس المرض. أول نوع من السكتة الدماغية يحدث عندما يتم قطع إمدادات الدم في لحظة ما إلى الدماغ يعرف بهجوم ischemic عابر .(TIA) الكثير من الناس يستخدمون عبارة "الضربة الدقيقة" لوصف هذه الحالة. وعلى النقيض من السكتة الدماغية النزفية الأكثر خطورة، فإن هذا النوع من الضربات كثيرا ما يكون قابلا للمعالجة.

تخطيط كهربية الدماغ (EEG) هو أسلوب سلسلة زمنية يستخدم كمثال في هذه الدراسة، في حين أن التصوير بالرنين المغناطيسي (MRI) هو مثال على مسار معالجة الصور. هناك إيجابيات وسلبيات لكلا النهجين؛ ولكن نظرًا للتوافر الواسع النطاق لمعدات ورقائق تخطيط كهربية الدماغ (EEG)، يمكن استخدام النوع الأول (EEG) في أي دولة، بينما لا يمكن استخدام جهاز التصوير بالرنين المغناطيسي (MRI). أكبر في الحجم والتعقيد ومتطلبات الصيانة.

استخدام أدوات أحدث مثل Python 3 وعدة خوارزميات مرتبطة مع تخطيط الكتلة المقترح في الفصل ٣، هذا العمل يستخدم استراتيجية ذات شقين لكشف السكتات من خلال EEEG ومعالجة الصورة. ويرد فيما بعد وصف تفصيلي لتدفق الإجراء.

تعتبر خوارزمية الغابة العشوائية هي الأفضل وتحصل على نتيجة أكثر دقة (٩٧%) وهي قيمة عالية. وفي طريقة IP فإن الإجراء المقترح الذي تم شرحه في الفصل الثالث والمذكور في الفصل الرابع حصل على نتيجة جيدة كما هو مبين في الجدول ٤,٥ وحصل على ٩٨,١% من ١٠٠٠ نتيجة. صحيح.

تم جمع مجموعة البيانات من مصدرين متميزين: مستشفى الهلال ومستشفى الإمام الصادق، اللذين تم استخدامهما خصيصًا مع مسار IP لصور التصوير بالرنين المغناطيسي؛ وموقع

Google المذكور في الفصل الثالث مع مزيد من الشرح، والذي تم استخدامه مع مسار مخطط كهربية الدماغ (EEG). قدم هذان المستشفيان مساهمات في تجميع مجموعة البيانات.

تذكر أن كلتا الطريقتين استخدمتا لغة بايثون، والتعلم الآلي، والخوارزميات المتخصصة. يجب أن توفر هذه الطرق الكشف عن السكتة الدماغية. وطالما أنها تعمل، يمكن إضافة خوارزميات وأساليب إضافية إلى عملية التطوير

شكر وتقدير

أتقدم بالشكر والعرفان إلى الأساتذة كل من الدكتور قاسم كرم عبد الله والدكتورة فرح نبيل عباس لإشرافهما على الأطروحة وتقديمهما يد العون والمساعدة في كل المجالات. كما يسرني ان اتقدم بالشكر والأمتنان لعمادة كلية الهندسة وقسم الهندسة الكهربائية للمساعدة في اكمال العمل

الباحث

بسم الله الرحمن الرحيم

سورة القلم – الاية- ١

إقرار لجنة المناقشة

نحن أعضاء لجنة المناقشة، نشهد بأننا أطلعنا على أطروحة الدكتوراه فلسفة الموسومة"(تصنيف إشارات الدماغ في السكتة الدماغية باستعمال تعلم الالة (إشارة الرنين المغناطيسي ورسم كهربائية الدماغ))

وقد ناقشنا الطالب (رياض عبد الحمزة محمد) في محتوياتها وفيما له علاقة بها، نؤيد أنها جديرة بالقبول لنيل درجة الدكتوراة فلسفة في الهندسة الكهربائية/ هندسة الألكترونيك والإتصالات

رئيس اللجنة
التوقيع:
الاسم: ا.د سمير جاسم محمد

عضو اللجنة
التوقيع :
الاسم : ا.د بيان مهدي صبار

عضو اللجنة
التوقيع:
الاسم: ا.د إبراهيم عبد الله مرداس

عضو اللجنة
التوقيع :
الاسم : ا.د سعد سفاح حسون

عضو اللجنة
التوقيع:
الاسم: ا.م.د مهند يحي ادريس

مصادقة رئيس القسم
التوقيع:
الأسم: أ.د. قيس كريم عمران
التاريخ:

مصادقة عميد الكلية
التوقيع:
الاسم ا.د. ليث علي عبد الرحيم
التاريخ:

اشهد ان اعداد هذه الأطروحة والموسومة ب

(تصنيف إشارات الدماغ في السكتة الدماغية باستعمال تعلم الالة (إشارة الرنين المغناطيسي ورسم كهربائية الدماغ))

(والمقدمة من قبل الطالب) **رياض عبد الحمزة محمد العلواني** (جرت تحت اشرافنا في قسم الهندسة الكهربائية/ كلية الهندسة/ جامعة بابل وهي جزء من متطلبات نيل درجة الدكتوراه في هندسة الألكترونيك والاتصالات

.

التوقيع :
المشرف الأول : أ. د. قاسم كرم عبد الله
التاريخ: / / ٢٠٢٤

التوقيع:
المشرف التاني : ا.د فرح نبيل عباس
التاريخ: / / ٢٠٢٤

اشهد ان هذه الأطروحة المذكورة اعلاه قد استكملت في الهندسة الكهربائية في كلية الهندسة

جامعة بابل /
التوقيع:
رئيس القسم: أ. د. قيس كريم عمران
التاريخ: / / ٢٠٢٤

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة بابل
كلية الهندسة
قسم الهندسة الكهربائية

تصنيف إشارات الدماغ في السكتة الدماغية باستعمال تعلم الالة (إشارة الرنين المغناطيسي ورسم كهربائية الدماغ)

الاطروحة

مقدمة إلى كلية الهندسة / جامعة بابل
كجزء من متطلبات الحصول على درجة الدكتوراه
في الهندسة الكهربائية/ الالكترونيك والاتصالات

من قبل

رياض عبد الحمزة محمد العلواني

بأشراف

الاستاذ الدكتور قاسم كرم عبدالله   الأستاذة الدكتورة فرح نبيل عباس

٢٠٢٤ م   ١٤٤٥هـ