Arab American University

Faculty of Graduate Studies

Detection and Classification of Abnormality in Electroencephalogram Signals Using Deep Learning and Convolutional Neural Networks

By

Kareem Tayseer Sadeq Khaleel

Supervisor

Prof. Dr Mohammed Awad

This Thesis was submitted in Partial Fulfillment of the Requirements for the Master's Degree in Computer Science.

February 2022

Detection and Classification of Abnormality in Electroencephalogram Signals Using Deep Learning and Convolutional Neural Networks

By

Kareem Tayseer Sadeq Khaleel

This thesis was defended successfully on **13/02/2022** and approved by:

Committee Members	Signature
1. Supervisor: **Prof. Dr Mohammed Awad**	
2. Internal Examiner: **Dr. Rami Hadrob**	
3. External Examiner: **Dr. Eman Droubi**	....

Declaration

I declare that this thesis entitled "Detection and Classification of Abnormality in Electroencephalogram Signals Using Deep Learning and Convolutional Neural Networks" is my work and has been composed solely by myself and does not contain work from others researcher and has not been submitted for any other degree or scientific except the reference is made.

Kareem Khaled

14/6/2022

Dedication

I dedicate this research to my family, friends, and all the ones who provided me with their unconditional love. To my mother who is no longer around me when I needed her the most. To My advisor, Prof. Dr Mohammed Awad who provided me with all the needed information to finish this work. For my friend Abedelkareem Zidan. And for the one who made all of this possible, to my wife Baraa. Thank you all for the unconditional love and support.

Kareem Tayseer Sadeq Khaleel

Acknowledgments

I would like to take this opportunity to express my deep regards to Prof. Mohammed Awad for his advice, support, and time that he spent reviewing my work. Prof. Mohammed provided valuable suggestions that have had a significant impact and helped in overcoming many obstaclesin writing this thesis in the best way.

Abstract

The human body is combined of a group of subsystems that communicate with each other, where the brain is the control unit for these systems. The brain function through a group of small parts called neurons, groups of neurons is connected to create a neural network to perform some tasks. When these neurons have issues or give wrong signals the brain will start to confuse the body systems and that can lead to neurological problems such as Epilepsy. To understand and diagnose such issues a test called Electroencephalogram (EEG) is performed. It records the electrical activities of the brain so that each part of this record can refer to an activity or group of activities. The issue with EEG is that it produces a huge amount of data for a short recording time, and due to the complexity of the brain signals and human errors, doctors are taking lots of time to diagnose the records, and many patients are misdiagnosed. This leads to the need to have a computerized system that can reduce these problems.

Many systems were proposed in the previous years, and with the advantages of AI and Machine Learning, much research was done in this field. Some applications were created using rule-based systems, others using Multilayer Perceptron (MLPs). When Deep Learning Networks such as Convolutional Neural Networks (CNNs) starts to be popular, many applications also were built based on them it. One main challenge of EEG signal processing is that the patterns are not necessarily unique; where the same signal for different patients can mean different things, which makes it very hard to create a generic model for EEG signal processing.

In this research, a new approach is proposed where we take advantage of the CNN abilities to extract features and handle complex time-series signals, combine it with wavelet signal decomposition along with preprocessing steps to create a robust model to analyze the EEG data and detect the epileptic seizures. The main strength of this work is that it's built at the patient level; since the EEG test provides a huge amount of data it's possible to tune the model for each patient. In this work, a 1D CNN model with Conv1D, Long Short-Term Memory (LSTM), and MLPS layers was created. A global dataset for several patients who are suffering from epilepsy was used. In this research, a comparison between our work and other regular algorithms such as Regression Tree, K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Ensemble was done. The algorithms we selected were based on the data structure and the studies that were done in this field. The proposed preprocessing algorithm enhances the data and made it easier to analyze, whereas the proposed detection model provided much better detecting accuracy when it's used with the processed data. Also, the regular algorithms did so based on the previous studies that were reviewed. The final average epileptic signals detection accuracy is 97.14%, with the highest accuracy of 99.2%. Originally this research was targeting the local Palestinian data, unfortunately, such kind of data was not available and due to that, a global data set was used.

Table of Contents

List of Figures

List of Tables

List of Equations

List of Abbreviations

EEG	Electroencephalogram
AI	Artificial Intelligence
NNs	Neural Networks
SVM	Support Vector Machine
CNNs	Convolutional Neural Networks
TP	True Positive
FP	False-Positive
FN	False-Negative
TN	True Negative
KNN	k-nearest neighbours
LSTM	Long short-term memory
HZ	Hertz
ILAE	The International League Against Epilepsy
CHB	Children's Hospital Boston

Chapter 1

Introduction

1.1 Introduction

The human body is composed of multiple subsystems that act and work together to perform all the needed actions. The main controller for these systems is the brain, which makes it the most important and the most complex organ of the human system. The complexity and structure of the human brain have attracted scientists for a very long time ago to study it and try to understand how everything works in it. Many researches have been done and many secrets had been reviled. However, there are still a huge number of things that we still don't know about. And that's why the field of human brain study is a very attractive topic for scientists.

In general, Artificial Intelligence (AI) algorithms are methods that try to mimic human brain actions. The baseline of the artificial neural networks is the neuron, which is built according to the actual neuron in the brain, where it's possible to mimic parts of the actual neurons using some mathematical models [1]. Figure 1.1.1 shows the main parts of the human brain: the spinal cord, brainstem, and large brain.

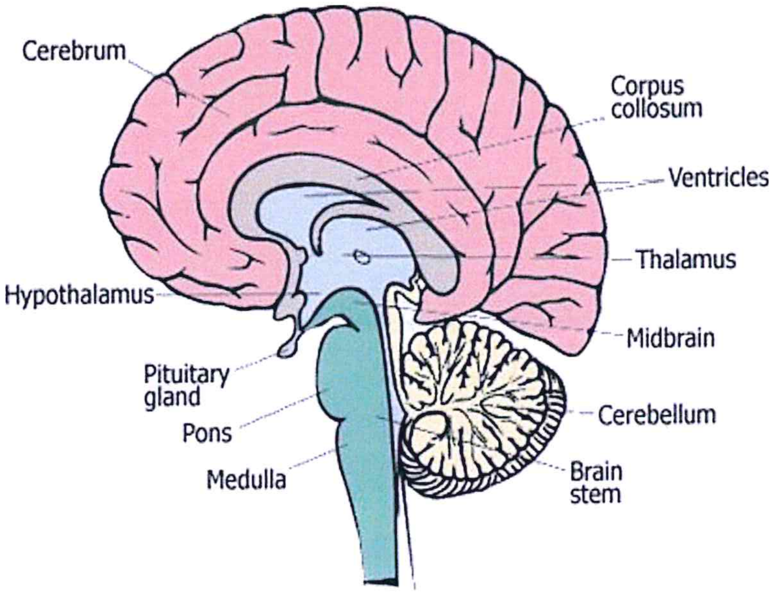

Figure 0.1: Parts of the Human Brain [2]

In the human brain, there are about 86 billion neurons connected to complex networks that control human actions [3]. One neuron sure will not be able to handle real or complex problems and will not be able to perform the needed actions. To be able to use the artificial neurons we need to have a network of neurons connected to increase the level of complexity and be able to deal with the real problems.

Electroencephalography (EEG) is one of the popular ways to record the analyze brain activities, that's because it is very easy to use, effective, doesn't cost much, and gives a good data resolution. When an activity occurs in the brain, neurons get activated, and that causes an electrical potential to be higher than the others, and that is the action that will be recorded by the electrodes attached to the scalp of the patient. The values and measurements will be different for each activity depending on the number of neurons that get activated, and their places [4]. EEG is considered one of the most important brain record tests due to its ability to record different activities happening within milliseconds. Also, EEG records are a deep insight into the inner situation of the patient's brain [5]. On the other hand, EEG records are very sensitive to electrical noise and can be affected by any electrical field, which makes them not reliable for real-world cases, so they are done in labs with a special environment. EEG is recorded by placing a headset containing multiple electrodes at specific parts of the scalp, the electrodes will measure the electrical activities inside the scalp. The output of the recording process is an EEG record with multichannel. The number of channels and values depend on both the number of electrodes and the reference of measurement [6]. Figure 1.1.2 shows an example of scalp electrodes placement, and what the output record will look like.

One of the most common neurological problems is Epilepsy; which is a neurological disorder that causes episodes of sensory disturbance, loss of consciousness, or convolutions. It also shows abnormal brain activities [7]. Epilepsy is considered the fourth most popular disorder that affects the human brain of all ages. It has many seizure types, and the level of control over the seizures can differ from one patient to another [8].

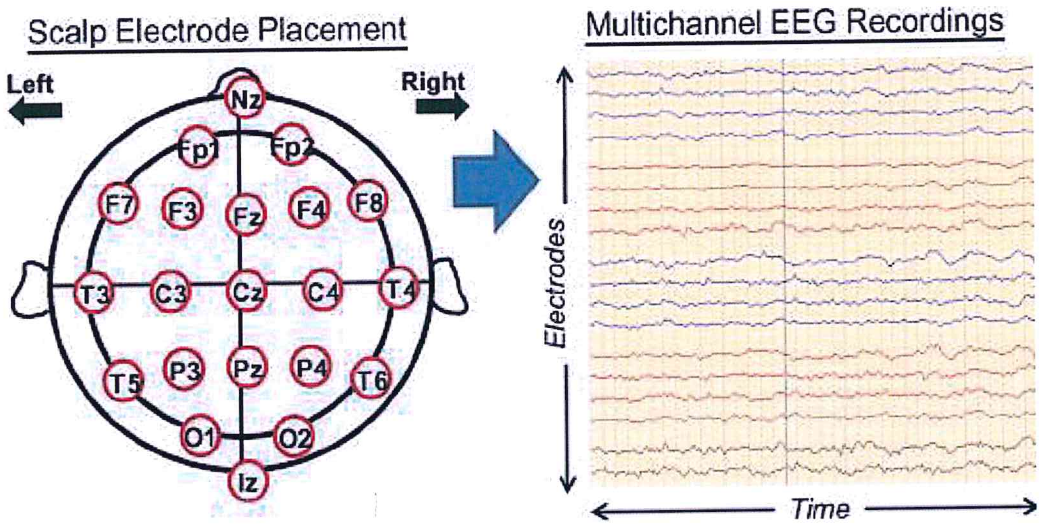

Figure 0.2: Scalp electrode placement and EEG output record [9]

The detection and analyses of epilepsy are done through EEG records. In many cases, it's hard to read the EEG and detect all the abnormalities in the record, which makes it essential to have better ways to handle the records especially when it comes to long-time records. Many computerized applications were created to help in this problem, some of them were basic, while others use very advanced methods including machine learning and deep learning.

Scientific and technological advances allow a broader understanding of neurological disorders and the various alternatives for their detection, classification, and prediction through the analysis of the electroencephalogram (EEG). In this sense, artificial intelligence (AI) techniques as well as machine learning techniques (ML) establish complex relationships between variables, providing models with important classifying and predictive properties based on the structure of the data [10]. These intelligent mathematical models are based on learning, which simplifies the classification process by pattern recognition.

Research shows that machine learning and deep learning techniques allow the detection and classification of EEG signals [11]. These techniques can detect epileptic using EEG signals using line length characteristics based on multi-resolution decomposition on the wavelet transform, which is combined with machine learning and deep learning to classify EEG signals about the existence or non-existence of a seizure. The objective of this thesis is to implement machine learning and deep learning techniques in EEG analysis to support medical diagnosis, develop a model based on CNN for the classification of neurological disorders through an analysis of real data and check evidence that CNN is indeed capable of classification. The model will be built at the patient level, which will give the ability to be more specific and more accurate. Also, the proposed model will take into consideration having simple architecture to make it easier to learn and classify so that it can be converted into a real-life application.

In this thesis, we will be comparing our model to a group of algorithms that are usually used for classification and especially the ones used for medical data such as Regression Tress, SVM, KNN, and Ensemble algorithms.

1.2 Objectives

In this work, the main focus is on the processing of electroencephalographic signals through the use of the Wavelet Transform and, on the ability to find mechanisms that allow automating the classification of the information obtained from such processing. The general idea of this research is to apply advanced machine learning methods to EEG data related to epilepsy disorder to create a model that can analyze the record, and classify the epileptic seizures.

Following are the general objectives of this research:

- Show how the Wavelet Transform allows obtaining quantitative variables that characterize brain electrical activity.
- Show, through the proposed analysis, that it is possible to capture the differences existing in the brain activity of epileptic seizures.
- Analyze the use of convolutional neural networks (CNN) in detection and classify the abnormality in Electroencephalogram Signals.
- Compares to CNN results with some of the most widely used methods in EEG analysis, such as K-Nearest- Neighbor (KNN), Regression Tree, Support Vector Machine (SVM),

and Ensemble Algorithm.

- Automatically quantifying and classifying the underlying information in brain signals constitutes the early diagnosis the epileptic seizures.

1.3 Detecting Epilepsy

Epilepsy is a neurological problem that can be pointed by unexpected cases of disturbance in senses, or unconsciousness, related to abnormal activities in the brain. *"The International League Against Epilepsy (ILAE)"* says that if any of the following conditions present, then epilepsy can be considered a brain disease [12]:

1. Two or more reflex or unprovoked seizures happen at about 24 hours intervals.
2. A reflex or unprovoked seizure and at least 60% chance of seizures similar to the general epileptic reflexes after detecting two unprovoked seizures in the next ten years.

One of the most common procedures used at the neurophysiology clinics to record the presence or evidence of epilepsy is the routine scalp electroencephalogram (rsEEG). It records the signals in epileptiform transients or ETs. The ETs that show spikes usually have a duration of 20 – 70ms, while sharp waves are 70 – 200ms, however, some come with slow waves present from 150 – 350ms. It is called a spike-and-slow-wave complex or usually called Generalized 3Hz Spike and Wave complex [13].

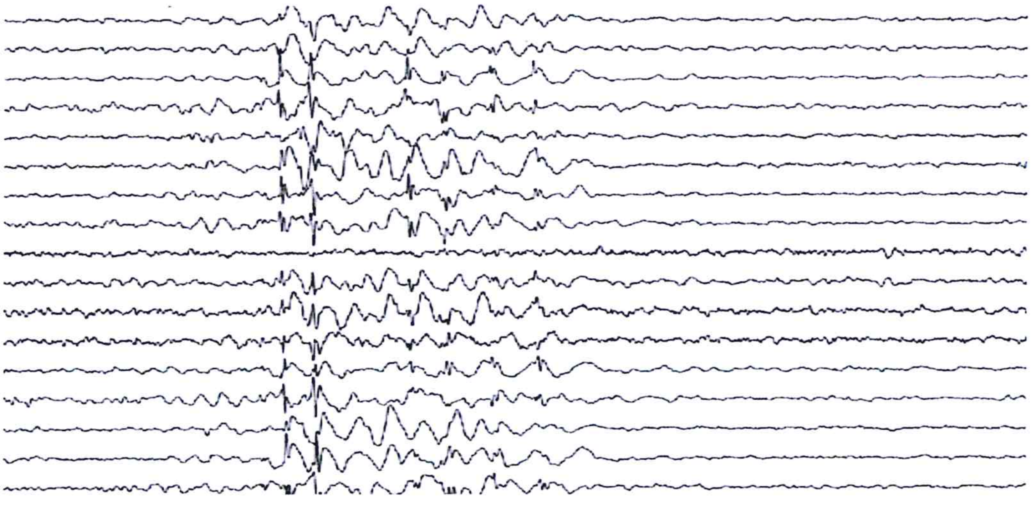

Figure 0.1:Generalized 3Hz Spike and Wave complex [14]

The shapes and appearances of ETs are very different, and they can be mixed with normal activities in the brain, so the detecting of ETs is not a direct process and it needs more work [15]. Diagnosing epilepsy is very hard due to the complexity of the signals. Human errors occur, and around 20 – 30% of the cases that are diagnosed with epilepsy are due to the similarity between the ETs and normal activities [16]. This shows that a patient can be misdiagnosed with epilepsy and take lots of unnecessary drugs and medications for many years for no reason. From all of the above, it's very clear that automated systems that can detect ETs more accurately are a must and especially advanced ones with Machine Learning Techniques.

1.4 Contribution

In this thesis, a case-level approach to detect epilepsy at the level of the patient using a dataset extracted from several patients who are suffering from epilepsy will be presented. An approach depending on training the algorithm at each patient data separately is followed since EEG records create a huge amount of data, which is sufficient to be used for training and evaluation. Also, a preprocessing method for the EEG raw data that will reduce the complexity of the data is proposed. The results had shown that when it comes to classification, the proposed model was able to provide higher accuracy than most similar methods. The preprocessing algorithm was able to reduce the level of complexity and help the model in producing high accuracy. These assumptions were tested by running the same CNN model with both preprocessed and raw data. This research will open the door for patient-level models, which will allow for better classification models and even can help in predicting future seizures.

1.5 Research Limitations

1.5.1 National Dataset

One of this research's main aims was to collect a Palestinian dataset. For more than a year and a half, it is tried to reach hospitals and clinics to collect the data. However, most of the cases were outpatients, and there are no real records that can be used. When the records are available (only 2 records with a length of 15 minutes, each one has about a 5-seconds seizure), the data was printed as an image and there are no digital copies of any kind for them.

One hospital located in Jerusalem has data similar to what we need. After contacting them for more than seven months, they accepted to provide us with part of the data, but they ask us to go there personally which was very hard, and they ask for permission to prevent us from publishing the research, which was not acceptable from our side. That's what force us to use only the global dataset, which was even hard to find a suitable one, because this kind of data is difficult to obtain, and most of the datasets were requiring paid subscriptions with a large amount of money.

1.5.2 Computational Power

Another limitation was the computational power required for the model to be executed. For example; the time to run the CNN for 10 patients is around 12 hours, without the time needed to do the preprocessing. This made the testing for multiple cases and multiple changes very hard, and most of this research time was spent on it.

1.6 Overview

The structure of this research is arranged as the following. In Chapter 2, We will talk about the background and the source of the EEG problem. Also, the information about the dataset used in this work is presented. Then we explain what are the types of signals and their meanings.

The next Chapter is 3, where we express in detail the model that was created with all the steps from preprocessing to network structure and how everything works. Also, the algorithms that are being used in this work will be explained. Chapter 4 will show the results of the research; where a comparison based on the different metrics is done. The conclusion and future work are illustrated. And finally, the appendix provided parts of the code that was used in this study.

Chapter 2

Background

2.1 Background

EEG records are hard to process and most of them are recorded as outpatients for diagnosis purposes. Most of the activities that are related to neurological issues are hard to be predicted and sometimes hard to be reproduced when needed, that's why most of the records from the clinics are not covering the problems and special facilities are required to have the patients connected to the recording devices for a long time so that they can capture the abnormalities and the correct brain activities. Epilepsy is one of the most popular neuro disorders which have a huge effect on the patient since it can lead to involuntary movements which can cause physical harm to the patient. That's why many studies were interested in recording and analyzing epileptic signals. When it comes to abnormal activities of the brain signals, most of the time they are related to neural problems, and the patients can't control them.

Based on the studies done on the epileptic patients, it was clear that the young ones are more affected by the seizures and in most cases, they can control themselves and which could result in serious injuries. Due to that, much research had been done in this field, and some facilities start recording a long-term record for the patients to allow researchers to have the needed data so that they can understand the problem and study it. In this research, deep learning with the CNN model is being used to detect and classify epileptic seizures in an EEG record. So, the following parts of this chapter will explain the dataset information that was used, and review what others have done in this field.

2.2 CHB Datasets Description [17][18]

Children's Hospital Boston is one of the most popular facilities that is concerned with neural problems, especially in children. They have provided a dataset for EEG records for children who are suffering from epilepsy. They provide multiple long records for groups of patients and provide the diagnoses where epileptic seizures occur. Records are grouped into 22 subjects; five males with the age range from 3-22 and seventeen females with the age range from 1.5-19. A file named "SUBJECT-INFO" is provided in the dataset, which contains the age and gender of each patient. The data for each patient (chb01, chb02, chb03, etc.) contains from 9 to 42 continuous records in the European Data Format (EDF). Most of the cases have exactly one hour or EEG recorded signal in the EDF files. However, some of them have two hours like chb10, and others have four hours of EEG data like chb04 and chb06. Usually, when the files that contain seizures detected are shorter.

The sampling rate that was used is the standard 16-bit resolution with 256 samples per second. The files in general have 23 EEG signals in each one of them, but some of them have 24 to 26 (depending on the number of electrodes). The International 10-20 electrodes placement system was used for these recordings (Figure 2.2.1). In some cases, there are additional signals recorded, like ECG, and VNS (Vagal nerve stimulus), however, these signals can be ignored in this work.

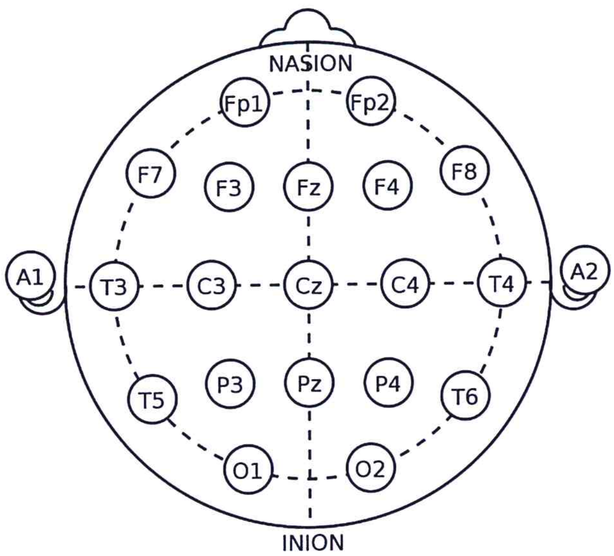

Figure 2.2.1: 10-20 system of EEG electrodes placement [19]

2.3 Types of Signals

EEG signals are complex signals that are defined by the rhythmic and transient. Figure 2.3.1 shows an example of a small part of an EEG signal. It's possible to extract features from the rhythmic based on the frequency bands. The EEG signal is very subjective to changes, for the same activity, the amplitude and frequency can be different from one person to another. Many factors can affect records like age, gender, and the status of the patient through the recording.

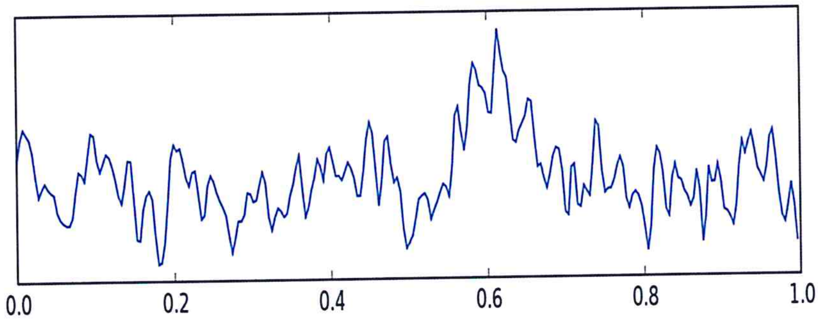

Figure 2.3.1: single EEG Signal [20]

Five types of bands can be identified and used based on the frequency analysis. They are alpha (α), theta (θ), beta, (β), delta (δ), and gamma (γ) where alpha is the lowest and gamma is the highest. A wave is usually produced by a specific part of the brain. However, this is not always true. Each mental state of a patient is associated with different waves. That can help to define the current state of the patient based on the waves recorded at a specific time. Table 2.3-1 shows the mental state related to each frequency band(wave):

Wave	**Frequency (Hz)**	**Mental State**
Delta (δ)	0 – 4	Deep Sleep
Theta (θ)	4 – 8	Drifting Thoughts, Dreams, Creativity
Alpha (α)	8 – 13	Calmness, Relaxation, Abstract Thinking
Beta (β)	13 – 30	Highly Focused, Highly Alertness
Gamma (γ)	> 30	Simultaneous Process, Multi-Tasking

Table 2.3-1: Frequency Bands and the Mental States

- **Delta Waves (δ)**

The range of the Delta waves is from 0 – 4 Hz. Usually, the mental state associated with delta can be deep sleep, comas, and hypnosis. It can also show in some cases when awake, however in this case it's a symptom of a neuro problem. The amplitude of the wave indicates how serious is the problem in a positive relationship.

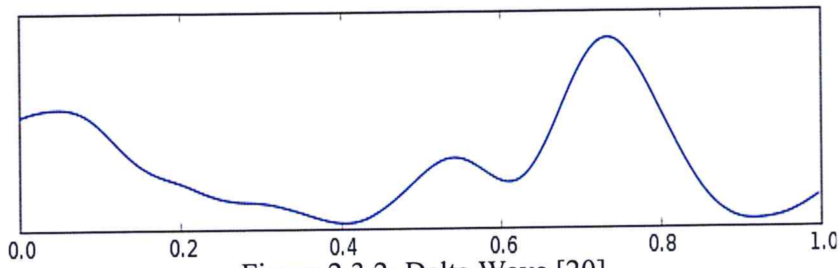

Figure 2.3.2: Delta Wave [20]

- **Theta Waves (θ)**

The range of the Theta waves is from 4 – 8 Hz. These waves appear usually when the person is deep thinking, has creative ideas, or has unconscious materials. Theta waves usually appear in healthy people in the phase of deep sleep.

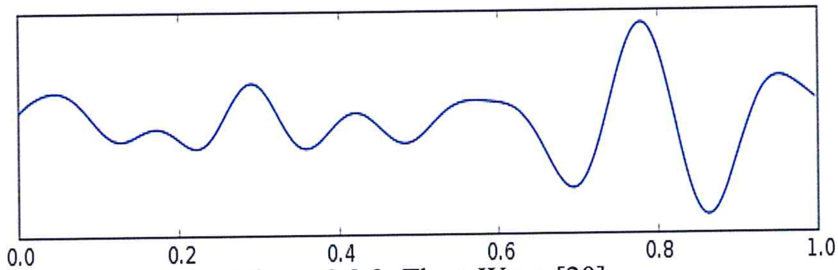

Figure 2.3.3: Theta Wave [20]

- **Alpha Waves (α)**

The range of Alpha waves is from 8 – 13 Hz. They appear when the person is relaxed and calm. Alpha waves usually have high amplitude in comparison to others. They can be captured when the person is awake and calm.

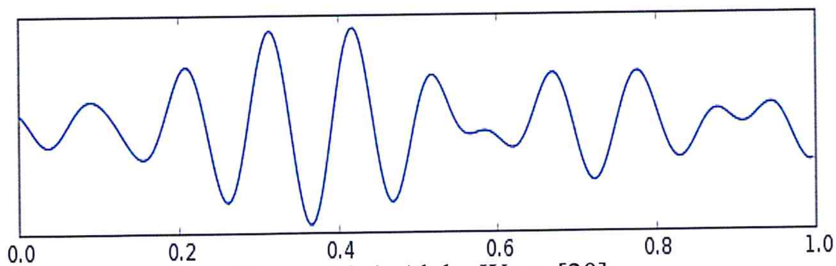

Figure 2.3.4: Alpha Wave [20]

- **Beta Waves (β)**

The range of Bata waves is from 13 – 30 Hz. They appear when the person is very focused and ready. The Beta waves range is large in comparison to others.

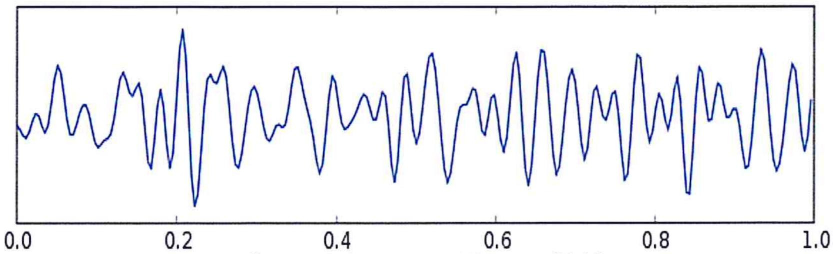

Figure 2.3.5: Beta Waves [20]

- **Gamma Waves (γ)**

The range for Gamma waves is higher than 30 Hz. They appear when the person is doing more than one task or doing simultaneous work. It's hard to notice the gamma waves, because their amplitude is very low, and they appear in all parts of the brain.

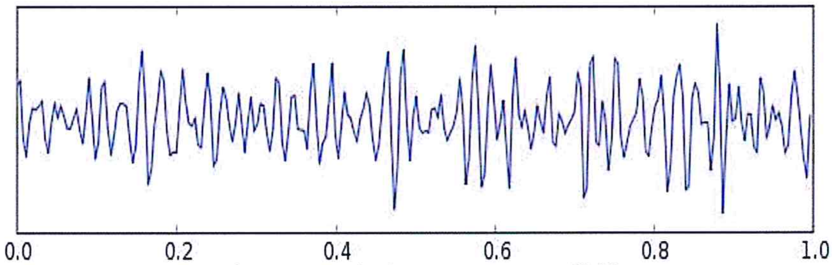

Figure 2.3.6: Gamma Waves [20]

2.4 Related Works

An early study is presented in [21], where the authors create a new algorithm for the medical EEG analysis for the detection of epileptic seizures. The work depends on making several changes to the EEG data itself to make it contains the spikes or signals related to epilepsy, and then classify it if they are epileptic or non-epileptic signals. The work done has one main problem, where they couldn't control the false positive detection, and depended on the electroencephalographer to detect these false detections. In [22], the authors created a new method that is patient-specific, where it reads a long time of recordings for each patient and analyzes it to detect epileptic seizures. The features were extracted from the time domain, and they used the nearest-neighbour algorithm to classify them. They manage to get a high result that reached 100% onset in some cases, with an average delay of 9.35 seconds, with a low false alarms average that reaches only 0.02/h. The limitation of this method is that it was trained on a specific templet, and it only can detect seizures similar to that templet.

The authors of [23] proposed a technique based on wavelet analyses to enhance the detection level and make it easier to handle the EEG data. They used the wavelet transformation to extract the frequency bands from the original EEG raw data. Although there weren't significant differences between the new extracted values, when they employed the parameters of one band through the functions, the differences appeared. The algorithm was tested on three groups: 1) healthy group; 2) epileptic group while no seizures; 3) epileptic group during seizures.

In [24], the authors propose to use non-linear dynamics tools with recurrent neural networks (RNNs). The main aim of the research is to check the accuracy of diagnosis on EEG when using RNNs with Lyapunov exponents using Levenberg–Marquardt training algorithm. They manage to reach an average accuracy of 96.79%, and according to them, this accuracy is higher than the regular MLPNN. In [25], a similar idea was proposed, where they used the non-linear algorithms with MLPNN.

In general, the work proposed by [25] (MLPNN), [26] (ANN), and [27] (RBFNN), share very similar ideas, where their proposed methods depend on the time-frequency analysis and developed algorithms that can detect the epileptic seizures in the brain. However, they all expect to have handcrafted feature extraction methods and manual changes to the EEG raw data. Although they provide a high detection accuracy (from 97.72% to 100% for [26] and 99.3% for [27]), these accuracies are limited to the circumstances and processing done by the authors and are not general results.

When it comes to features extraction, most of the work to detect the features and extract them is done manually by the experts, and then fed to the machine learning models. This is done very carefully since it's a very sensitive process, and if it was not done correctly, the results will not meet the expectations. To overcome the mentioned problem, many researchers tried to find an effective and automated method to extract the features of the EEG. The authors of [28] proposed a wavelet-derived approach to extract the EEG features. The approach proves that it's possible to enhance the classification performance. Hence, the research shows that the EEG classification accuracy did increase when using the proposed approach. They used K-NN with 10-fold-cross-validation to evaluate their work. In [29], a features method that is based on the power spectral density was proposed to enhance the detection of yellow boxing (a potential abnormal epileptic signal). The features are extracted as power values, and the sudden high-power changes in the signals are marked as a potential yellow box signal. According to the researchers, this method is effective and helped them to achieve high accuracy. In [30], the authors used Linear predictive coding (LPC) for feature extraction. LPC is a method used originally for speech signal processing, where it changes the input signal into a group of parameters. Since the shape of EEG signals is close to the shape of speech signals, it's possible to use LPC for EEG feature extraction. According to the work done by the authors, this method provided good results. In [31], the authors proposed a method where they used Radial Basis Function neural networks to detect Epileptiform transients (ETs) signals from the raw EEG signal. After that, they used the Hilbert–Huang transform (HHT) method to extract the features from the ETs to classify whether the ET is a real epileptic seizure or not. After analyzing the above methods, we decided to go with wavelet feature extraction since its reliable, accurate, easy to implement in real-life applications and can give fast results.

Authors of [11] provided one of the methods that started to use computer vision with deep learning and convolutional neural networks with an end-to-end learning method. The main idea of the presented work is to use create a visualization of the EEG data, and apply the features of deep learning to classify the EEG signals to detect epileptic seizures. In their work, they used FBCSP and ConvNets algorithms, and manage to get an accuracy of 82.1% and 84.0% respectively. Although the results are not very accurate, it's worth mentioning that they use raw EEG data, and let the deep learning algorithms do all the work. The main advantage of this research was the visualization methods they provide, which make it easier to read and understand the EEG data. Working with images means that the data shape will be in 2D, which can lead to complicating the structure of the neural network. To overcome that, the authors of [32] proposed an enhanced deep one-dimensional convolutional neural network that can detect abnormalities in the EEG signal. The proposed model is an end-to-end system without feature extraction. The study was done using a single-channel EEG signal (temporal to occipital T5-O1). The model provides classification with an error rate of 20.66%.

The research done by the authors of [33] uses the dataset [17] to create an automated seizure detection model which was implemented by using a Field-programmable gate array (FPGA). To enhance the model, the used wavelet-based algorithm will extract the features from the EEG and group them into 4 categories. They used the ant colony optimization technique to create a rule-based classification system and were able to get a testing result of 98% after testing it in another dataset. In [34], the authors mainly try to use wavelet decomposition to redefine the EEG signal morphology and spatial distribution to create a vector of features. After that, they used SVM to do the classification. The highest accuracy they can reach is 97%, however, this result was taking time and a have a latency of 9.1 seconds, which means that the system was able to detect the seizure after it happened with 9.1 seconds with the mentioned accuracy. In [35], a deep learning algorithm was proposed depending on the convolutional neural networks which they used to detect the seizures. They used another dataset (Bonn university dataset [36]), and their system can reach 88.67% accuracy, a sensitivity of 95%, and a specificity of 90%. In their work they consider a model of 13 layers CNN, having 5 convolutional layers, and instead of having two output classes, they have three.

In [37] the authors provide a software-based neural network that aims to detect seizures in raw EEG data. The software uses a convolutional neural network that was trained and evaluated on the raw data from the "CHB-MIT" [17] dataset. The author's expectation from this study is to create a portable system that can be exported to a chip (system on chip SoS). They manage to achieve an accuracy of 96.74%.

In this research, all the work done above was taken into consideration, and the main idea presented is to have a system with a simple structure, and with the principle of training the system at the patient level. A 1D CNN will be created with wavelet features extraction after a preprocessing algorithm and the "CHB-MIT" dataset will be used for training and evaluation. The system will be patient-specific, which means it will train on each patient's data separately. The final results are great and an average accuracy of 97.14% was achieved, whereas the highest accuracy for a single patient is 99.2%.

Chapter 3

Methodology

3.1 The Proposed Method

This chapter will demonstrate the aspects of the proposed model. We will go on to explain all the steps required to build this model, and what is the purpose of each step.

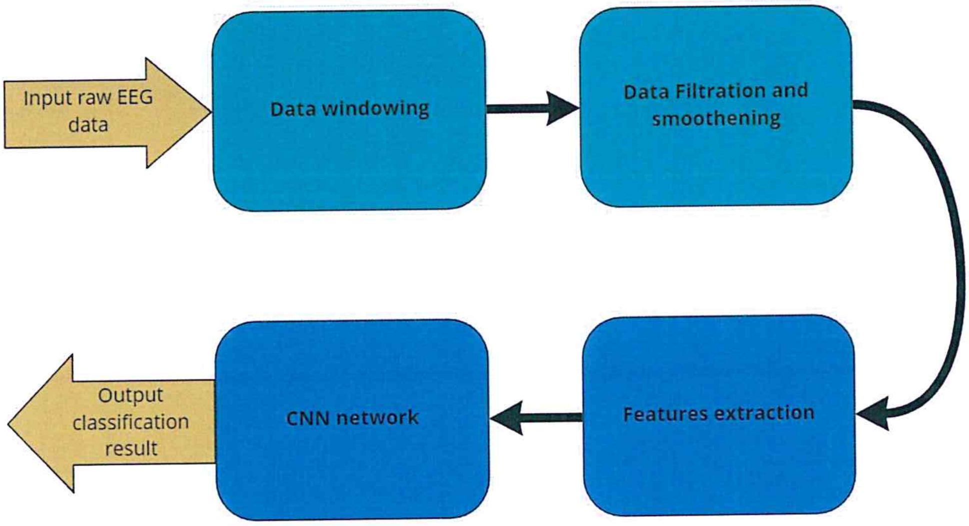

Figure 3.1.1: Main model structure

The EEG signals are very complicated and can be very different from one patient to another, they can be different for the same patient under different circumstances. Also, the signals representing a single brain activity can be sometimes very similar, or they can be different, however, they will share some main features.

Based on the previous research most of the work is done was aiming to find a general model that can detect or predict some actions of the brain, and in this case; detecting epileptic seizures. Although many of them succeeded in getting high accuracy, most of them are giving general answers, and in many cases, the algorithms are giving false results if the patients' signal patterns are new to the model. Also, the same pattern that is normal for a patient can be marked as abnormal for another one.

To make the above idea clearer, figure 3.1.2 show the starting of an epileptic seizure for a patient

(CHB01), and figure 3.1.3 also shows another seizure for the same patient. Although they seem to be similar, however, there are some differences in the wave's shapes, amplitude, and recurrency. Generally, machine learning techniques will be able to detect these features and a good model will be able to detect such a pattern. But when looking at figure 3.1.4 which represents the starting part of epileptic seizure for the different patients (CHB06), the differences start to be more noticeable, and the features that helped in detecting the seizures in the previous case become worthless and can give might causes false results. That leads to a better idea which is to have a model able to detect the seizures at the patient level, where the huge amount of data in a single EEG record allows the model to be trained, tested, and validated. And that will make the model better in detecting and analyzing epileptic seizures in EEG signals.

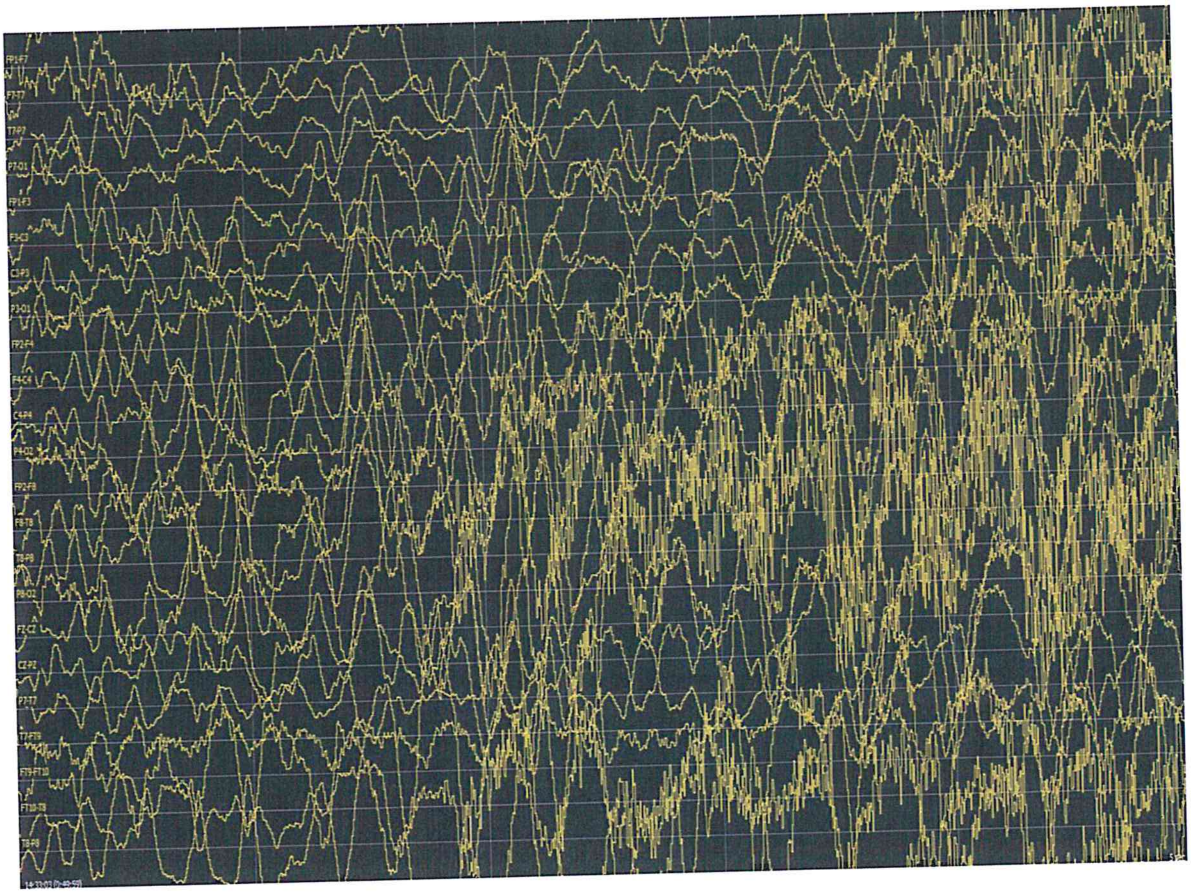

Figure 3.1.2: starting of an epileptic seizure for patient CHB01

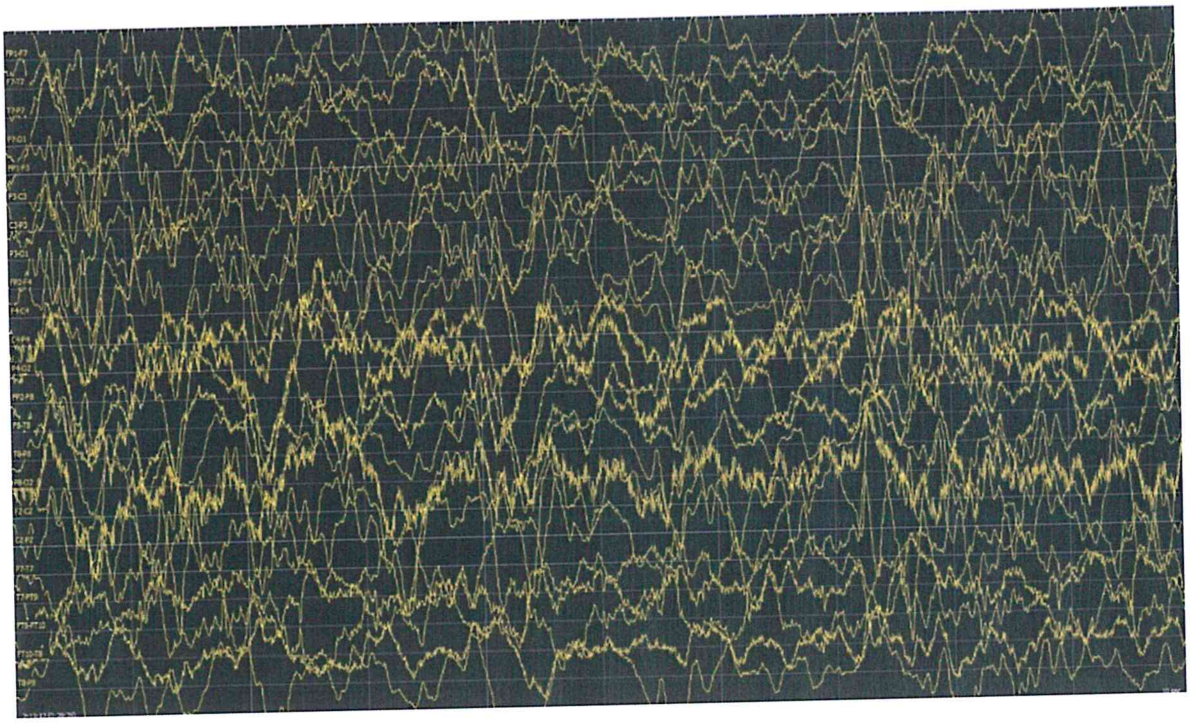

Figure 3.1.3: starting of another epileptic seizure for patient CHB01

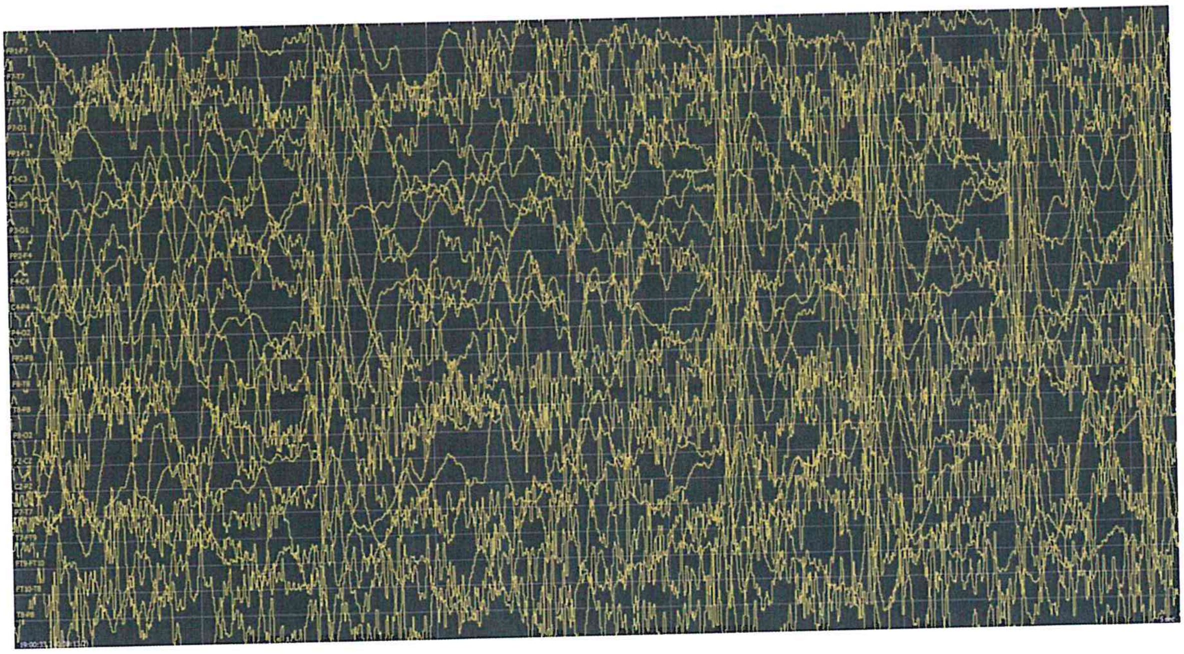

Figure 3.1.4: starting of an epileptic seizure for patient CHB06

The data presented above shows how it can be very hard to contain all epileptic features with a general model, and if it was done, the model can also give wrong answers sometimes. In this thesis, the model used for epileptic signals classification is built at the patient level. Since a single EEG record can present hundreds of thousands of data points, it will be possible to have enough data to be used for training and testing. Also, it was found that despite the mentioned differences in the records, the main EEG features for a single patient will be very similar. For example, the value of Gamma when there is an epileptic seizure is almost reaching the same peek for the same patient. However, it can change for a different patient.

Since the features will be similar, the detection will be easier and it will take less time to train and give the results. Also, such a model can have continuous training to be updated for the patient case, because the signal strength and features can change when the patient gets older, and this will give the model the ability to be applicable in real life and to be used in different ways such as diagnostic support system or patient status monitoring.

3.2 Data Pre-Processing and Preparation

EEG data is very complex and it's very subjective to changes, where a small action or movement can lead to high noise and unclear signals. Figure 3.2.1 shows a short recording of five seconds, and it's clear that it has a huge amount of data.

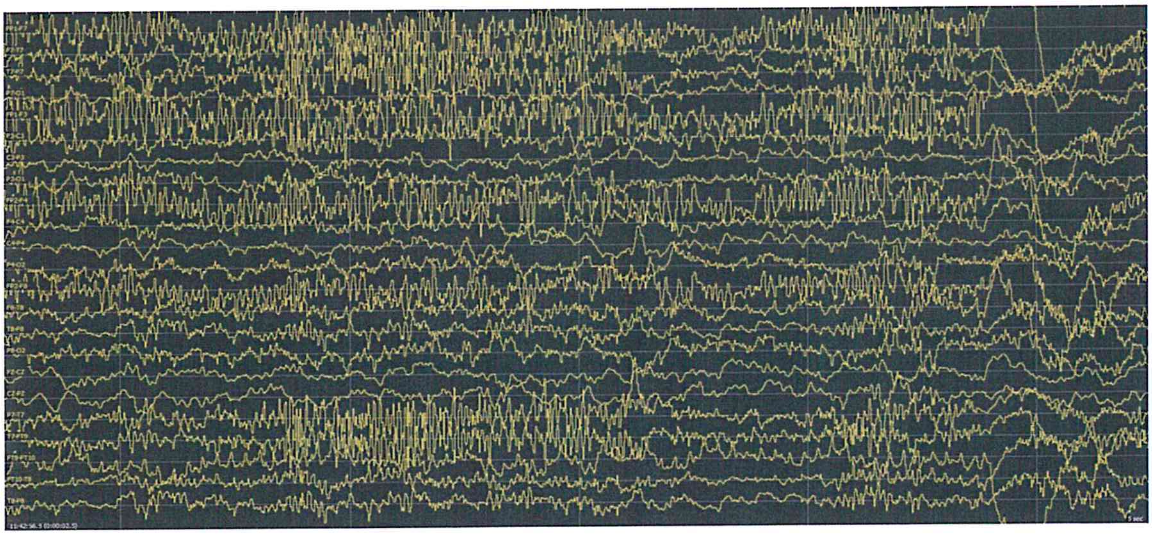

Figure 3.2.1: Real EEG signal showing 22 electrodes recording for 5 seconds

The records in the dataset are stored as an electrical value that represents the amplitude of the signal at a certain point in time. Each record has 22 rows of data, each row is sampled at 256 samples per second. Most of the records are one hour in length which produce about 921600 samples per row, multiplied by 22 rows for a single record. To be able to handle this huge amount of data, it's important to process it and get the parts that can be used to train the machine learning application. This process is important because for the one-hour record an epileptic seizure can last about only five seconds. And that will make the application inaccurate.

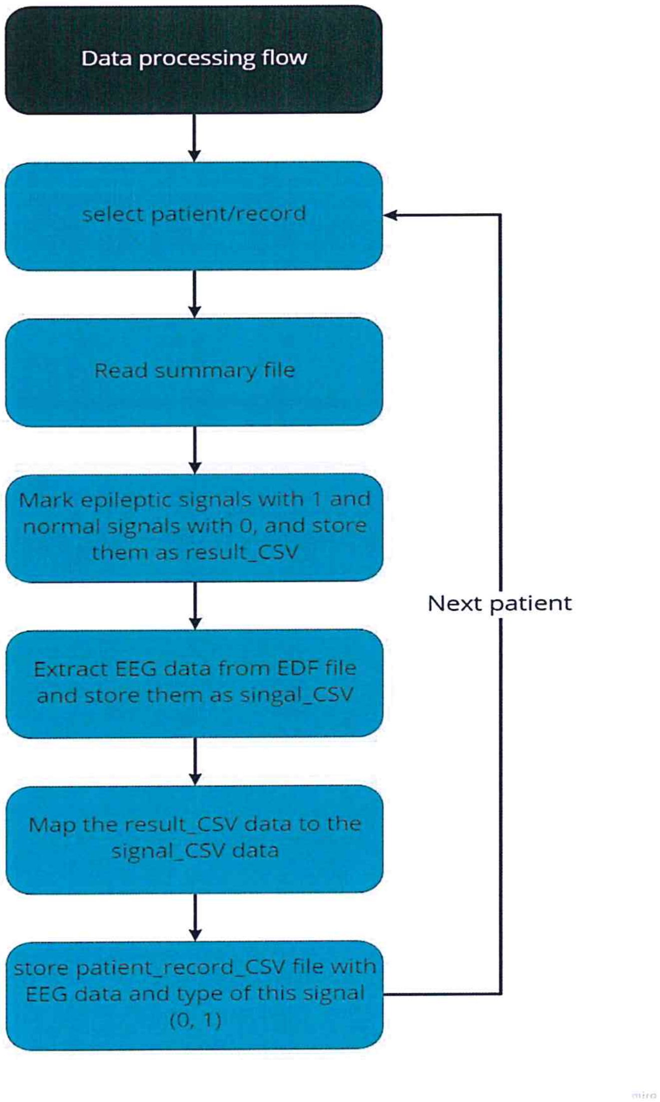

Figure 3.2.2: Initial data processing

Figure 3.2.2 presents the steps that were taken to process the initial dataset files:

1- Select the patient record: a patient has multiple records stored in EDF files. The files are initially processed to export the EEG readings as a 2-D array of time and electrodes values

2- Read Summary file: for each patient, there is a summary text file that gives the needed details about the patient, records and seizures.

3- Mapping steps: from the exported data from steps 1 and 2, the EEG data points that were pointed as epileptic from the summary file were marked with 1 and normal signals were marked with 0. Then a CSV file was exported containing both the EEG data and the status of each datapoint as normal or epileptic.

At the end of this phase, a file for each record was created that maps the results to the raw signal in CSV format.

3.2.1 Windowing

Since the size of the data produced by an EEG record is huge, it's important to find a way that can reduce the amount of data without harming the information.

Windowing is an algorithm that takes a series of points and averages the values to create one point that represents that group of points.

This technique will help in reducing the number of points and if the parameters were set correctly, the amount of information lost in this process will be minimal.

The min parameters that control the windowing process are:

1. Size: This parameter controls the length of the window in seconds. The actual size of the window is the size multiplied by the sampling rate.
2. Step: This parameter controls the next window starting point. By setting the step it's possible to have separated windows or to have overlap between the windows. If there is no overlapping, the information at the edge of the window can be lost or harmed. Overlapping will reduce the amount of missing information. However, it will increase the number of total output points
3. Function: This parameter provides the chance to create a custom function that runs on each

window data and returns the results. In general, the average function is used.

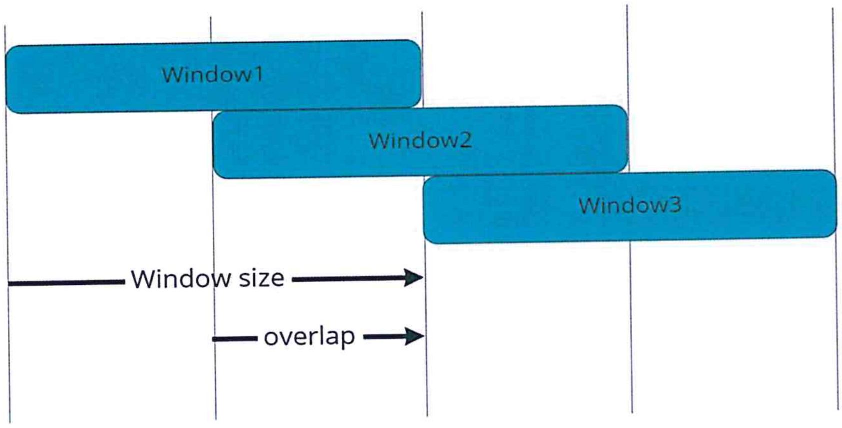

Figure 3.2.1.1: Windowing process showing overlapped windows

3.2.2 Data Filtration and Smoothing

The EEG test is recording very low power signals that are very sensitive to change. Any movement from the patient or the headset can cause the signal to jump in amplitude in a very sharp way causing a spark signal, and usually, these signals are not showing the real state of the brain and they should be handled.

To remove the sparks and noise without harming the original signal three methods were applied:

- Low pass filter: This filter is applied to the signal to remove the signals with a frequency higher than 40 Hz. This will help in removing the sparks and very high-frequency signals.

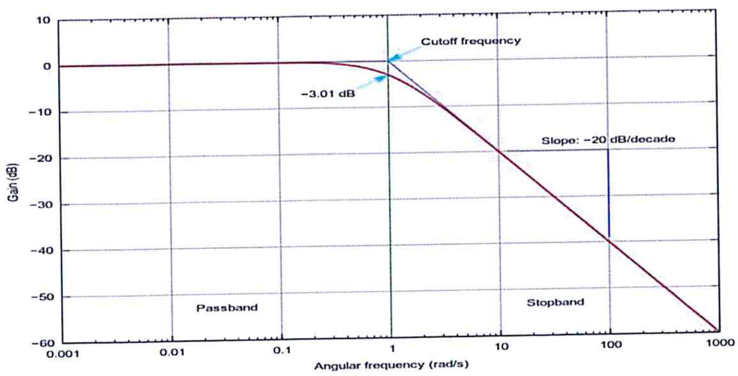

Figure 3.2.2.1: Low-pass filter of first-order

The following is the transfer function of the low-pass filter of the first-order with a cutoff frequency of 1kHz:

$$H(s) = \frac{\omega c}{s + \omega c} = \frac{1}{1 + {}^{s}/_{\omega c}}$$ EQ 3.2.2.1 Low pass filter

Where $\omega_c = 2\pi * 1000$

- High pass filter: This filter is applied to reduce the noise and remove the signals with a frequency of less than 4 Hz. It will help in reducing the complexity by removing the flapping from low-frequency activities and removing the noise.

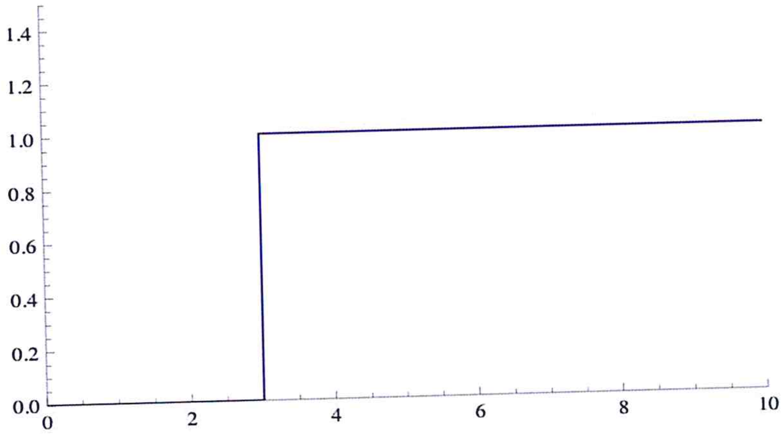

Figure 3.2.2.2: Ideal high-pass filter of first-order

The following is the transfer function of the high-pass filter of the first-order with a pass frequency of 2kHz:

$$H(s) = \frac{s}{s + \omega c} = \frac{s/\omega c}{1 + s/\omega c}$$ EQ 3.2.2.2: High pass filter

Where $\omega_c = 2\pi * 1000$

- Smoothing: This process is performed based on the idea of moving average where it takes N points, calculates their average, and replaces the original values with the average. Then it includes the next point and does the same process again. It is important to be aware that this process is not safe all the time, and it can cause damage to the data accuracy. However, it is a strong method to smooth the sparks and remove the nose.

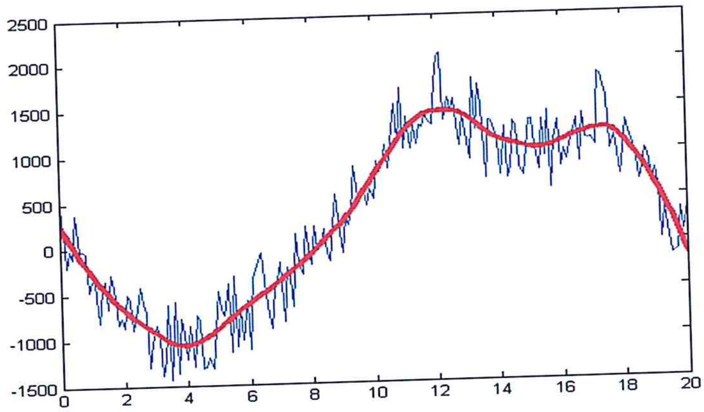

Figure 3.2.2.3: Smoothing output example

3.2.3 Wavelet Transformation

Fourier Transform (FT) is one of the most popular methods to transform the signals from one domain into another domain. Many researchers used FT as a signal decomposition to extract the information from EEG. However, one big issue with FT is that it captures the "*global frequency information*", which means that some frequencies will affect all the signals. The FT decomposition is not the ideal solution for signals with short intervals and oscillation features like the ones in EEG and ECG signals. One of the most popular alternatives for FT is the wavelet transformation which transforms a function into a group of wavelets [38].

Wavelets have two main features: location and scale. Location is the parameter that defined where does the wavelet position in time or space. Scale defines the structure of the wavelet; it controls where it is "squished" or "stretched".

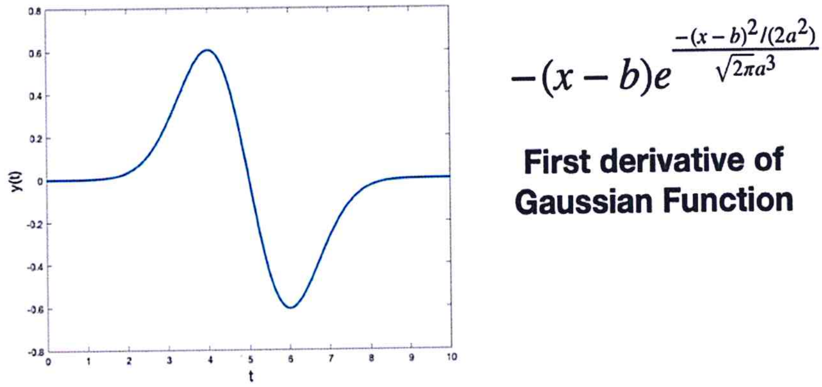

Figure 0.1: Example Wavelet

In the wavelet example in figure 3.2.3.1, the "a" parameter controls the scale of the wavelet. If the aim is to handle high-frequency information, "a" value is decreased, which will cause the wavelet to be more squished. In the contrast, if the aim is to handle low-frequency information, "a" value is increased, which will make the wavelet more stretched. The next figure shows both squished and stretched shapes of the same wavelet shown in figure 3.2.3.1.

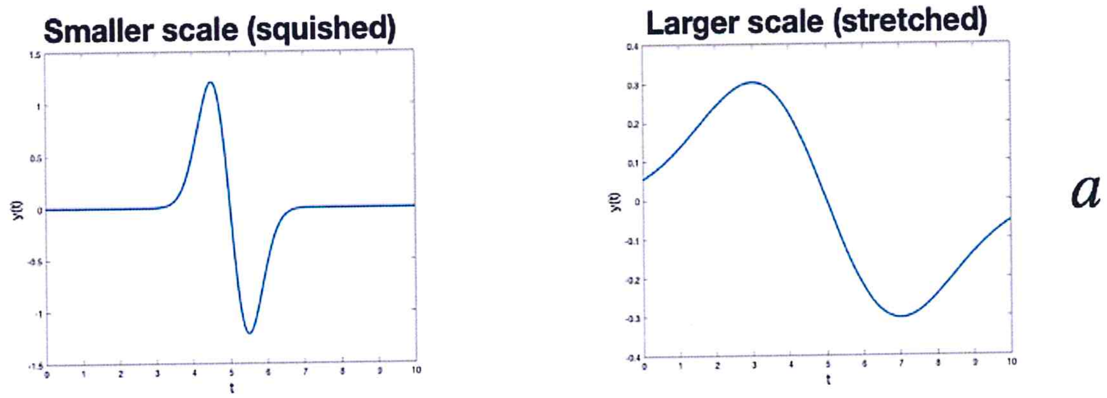

Figure 0.2: Left; wavelet with small "a". Right; wavelet with large "a"

The "b" parameter in the equation in figure 3.2.3.1 controls the location of the wavelet. The smaller value assign to "b", the more it will be shifted to the left, and the greater value assign to "b" the more shift to right. The location of the wavelet is very important because wavelets are not like waves, where they only have values for sort intervals. Also, when using wavelet decomposition, we are not only interested in the changes in the signal, but also in where did these changes take place. Figure 3.2.3.3 shows an example of how changes in "a" and "b" will change the wavelet shape.

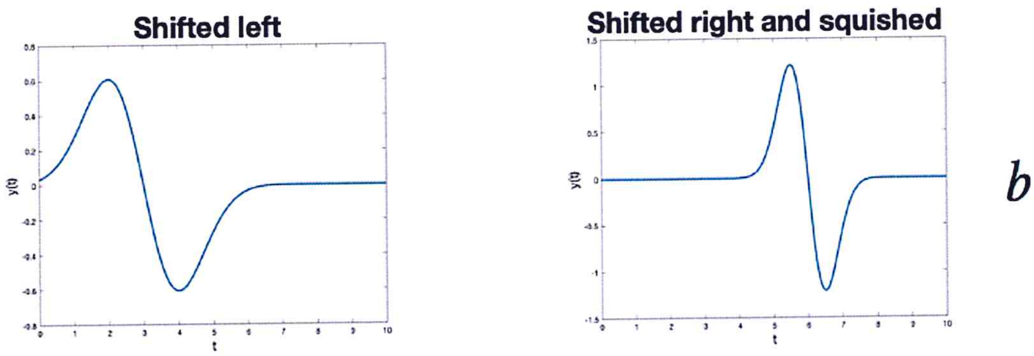

Figure 0.3: Left; wavelet with decreased location and increased scale.
Right; wavelet with increase location and decreased scale.

The idea of wavelet decomposition is to select a wavelet with a specific scale. Then slide the wavelet through the entire signal by changing its location (change the value of "b"). Every time step the wavelet is multiplied by the signal value. The result of the multiplication gives us the coefficients of the wavelet signal at that time. Once we are done, we change the scale (value of "a"), and repeat the process.

There are two main types of wavelet transformation: Continuous Wavelet Transformation (CWT), and Discrete Wavelet Transformation (DWT). The two types are very similar, and from figure 0.4, we can see the equations for CWT and DWT. However, the main difference between them is that CWT is done through all possible wavelets with a range of defined scales and locations. On the other hand, DWT uses a specific set of wavelets that are defined at a group of locations and scales.

Continuous Wavelet Transform (CWT)

$$T(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t)\, \psi^{*} \frac{(t-b)}{a} dt$$

Discrete Wavelet Transform (DWT)

$$T_{m,n} = \int_{-\infty}^{\infty} x(t)\, \psi_{m,n}(t)\, dt$$

Figure 0.5: CWT and DWT

Many wavelets' signals can be used, and this is one of the main strength points in WT, where you can choose the wavelet based on the problem and input signal type. Another main strength point is that WT can extract both local temporal information and spectral. Figure 0.6 shows a group of wavelet signals. In this research, db4 was selected.

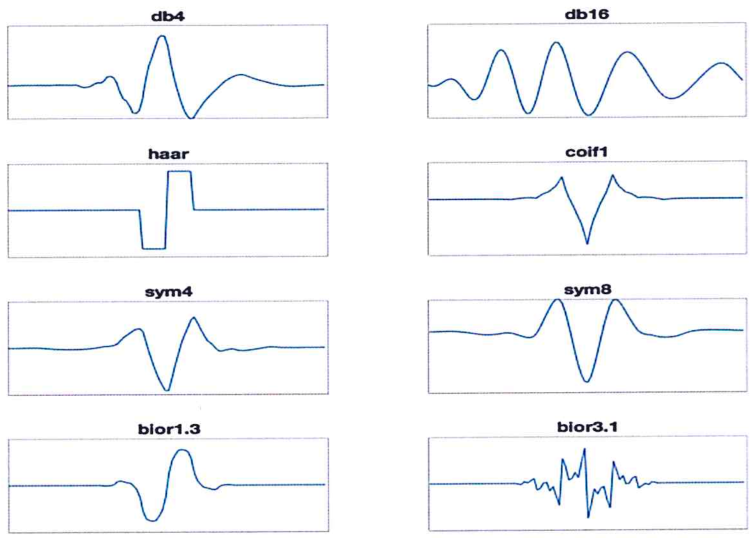

Figure 0.7: Different types of wavelet signals

3.3 Feature Extraction

To handle the information stored in big data and detect the patterns it's better to extract the features that can differentiate between the outputs.

In EEG data, there are main features that can be extracted to reduce the amount of data and can help the process to be done faster and more accurately. These features are called frequency bands, and they are explained in detail in section 2.3. Following is the process that was taken to extract the features.

- Apply lowpass filter, high pass filter, and smoothing function to remove all unnecessary frequencies.
- Apply a windowing algorithm with a window size of 25% of a second, and 0.5 overlaps between windows to make sure that all the edges are covered.
- Extract the frequency bands using wavelet transformation (Theta, low Alpha, high Alpha, Beta, and Gamma).

- Theta: average power in the 4 to 8 Hz frequency band
- low Alpha: average power in the 8 to 10 Hz frequency band
- high Alpha: average power in the 10 to 13 Hz frequency band
- Beta: average power in the 13 to 25 Hz frequency band
- Gamma: average power in the 25 to 40 Hz frequency band
- For each point in time, there will be 5 features multiplied by the number of electrodes.

- To reduce complexity an averaging function was applied at each time point to have only one frequency band value at that time.

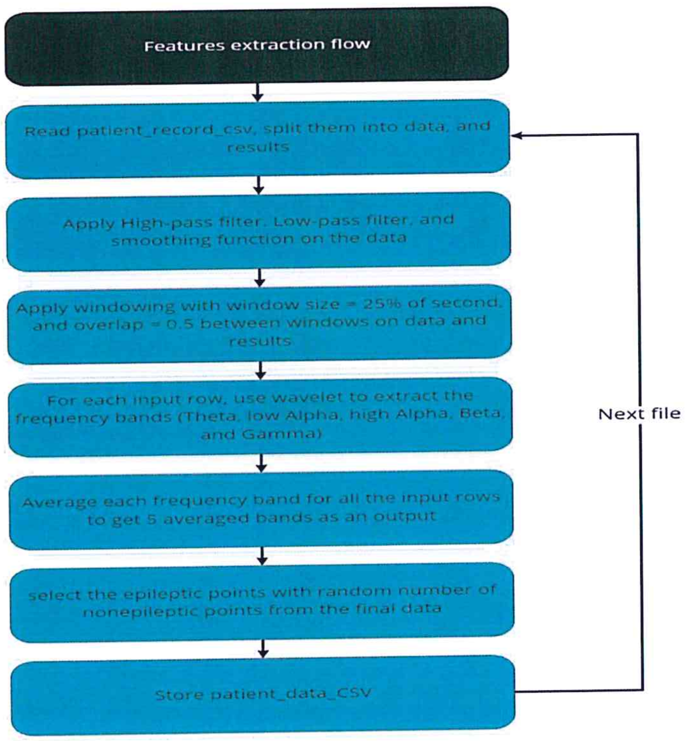

Figure 3.3.1: Proposed features extraction flow

This process extracts the important information from the EEG data, and instead of having complex long records, it provides more usable and smaller data points. Following a flowchart explains in detail the process of extracting features

3.4 Machine Learning Techniques to Classify EEG Data

The main idea of Machine Learning (ML) is to be able to use known data to identify and detect unknown data. There are many machine learning algorithms, each one has its pros and cons. The problem, data, and expected results are the main points that control the process of selecting an algorithm. Artificial Neural Network (ANN), Genetic Algorithm, and Support Vector Machine are some of the most commonly used algorithms to classify EEG data [39].

Many machine learning algorithms are available and can be used to analyze and classify EEG signals. However, some are being used more than others, and the following are the algorithms were chosen in this study:

- Regression Tree (RT)
- K-Nearest Neighbour (KNN)
- Support Vector Machine (SVM)
- Ensemble
- Convolutional Neural Networks (CNN)

The first four techniques are a set of the most popular techniques and are most widely used almost for all problems. Also, CNN is one of the new promising algorithms that can handle EEG data.

3.4.1 Regression Tree (RT):

The idea of the Regression Tree (RT) algorithm is to give a decision on the category or output of an input situation or object based on its properties [40]. The features are represented as nodes and the test possible results are the branches of that node. Once a node is reached, the tests are applied to it, and the final nodes are called terminal nodes; where they contain the results or the decision values. The level of the tree depends on the features and inputs.

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_j - y^{\wedge}{}_{R_j})$$ EQ 3.4.1: Euclidean distance

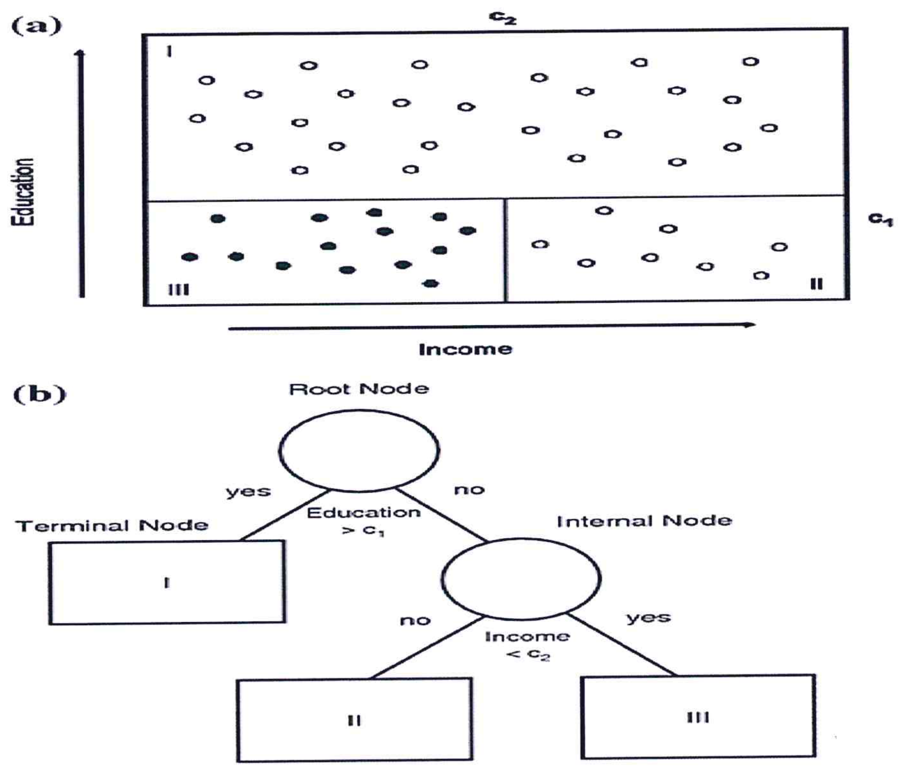

Figure 3.4.1: Example of Regression Tree [41]

The regression tree algorithm is very popular when it comes to the medical field, where it's been used widely in the classifications like heart attacks, speech recognition, and some diagnosis of cancer. [39] [40]. Following is a pseudocode implementation for the algorithm

```
1.  Initially all observations are in the root node;
2.  Start from the root node, partition the samples using the following
    recursive procedure;

3.  Giving a tree node;
    4.  Perform stepwise linear regression on samples in current node;
    5.  Calculate the residual sum square (RSS_node);
    6.  If (number of samples > predefined minimum node size)
        7.  For each predictor variable x_i
            8.  Sort x_i in ascending order;
            9.  For each value d in the above sorted list
                10. Using d as the threshold value, partition the samples
                    within current node into two subsets;
                11. Perform stepwise regression on each subset and calculate
                    the corresponding RSS;
                12. Calculate the subtotal RSS as the sum of those of the two
                    subsets;
            13. End of for loop on threshold value d;
            14. Find the minimum subtotal RSS achievable from splitting
                current node on x_i;
        15. End of for loop on predictor variable x_i;
        16. Find the minimum RSS of all possible splits of current node
            (MinRSS_splitting);
        17. Calculate the improvement from splitting the current node
            using equation (4)
        18. If (Improvement > predefined minimum improvement)
            19. Split current node into two new nodes using the variable
                and threshold value that give the MinRSS_splitting;
            20. For each of the two new nodes, go to step 3;
        21. End of if in step 18;
    22. End of if in step 6;
23. End of recursion.
```

Figure 3.4.2: Regression Tree Pseudocode [42]

3.4.2 K-Nearest Neighbor (KNN)

One of the simplest and basic algorithms used for classification is the K-Nearest Neighbour (KNN) [43]. In general, KNN is used when the data is ambiguous, and there is very little knowledge about how the data is distributed. This algorithm is very strong when it comes to the classification of problems with no parameters, and uses the probability densities to do the classification [44]. The process of classifying is done by assigning a value X as a label for the most repeated value among the K-nearest points. This will allow the algorithm to voting the labelled point as a reference value for classification. KNN will do the classification based on the distance between the point and the closest label [43].

$$y = \sqrt{\sum_{i=1}^{K}(x_i - y_i)^2}$$ EQ 3.4.2: KNN output class

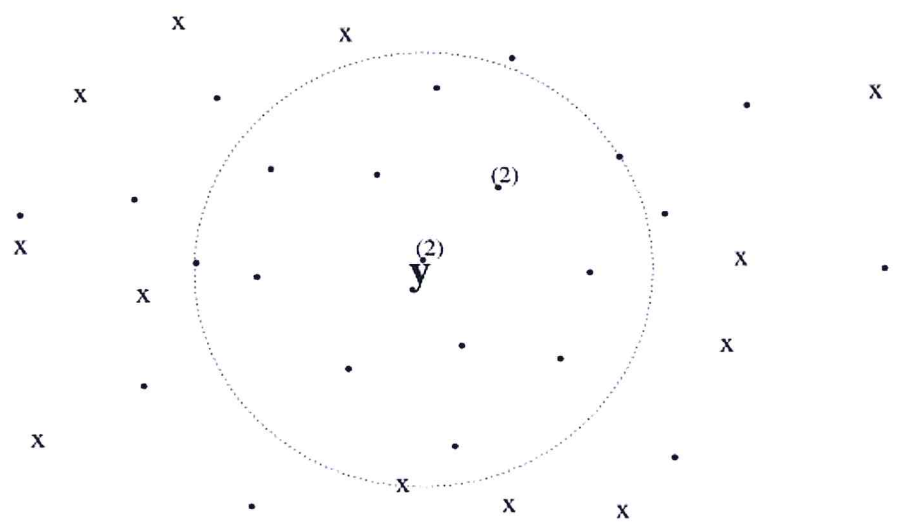

Figure 3.4.3: Classification using KNN [45]

Figure 3.4.3 shows an example of classification done by KNN where '.' represents points that share features with the point labelled as class 'y', and 'x' represents points with features of other classes rather than 'y' [45].

The following figure expresses one of the algorithms used to implement KNN

```
Input  :  D, a chunk of the original distance matrix;
          n_chunksize, dimension of the chunk;
          split, index of split;
          chunk, index of chunk;
          Maxk, an array to hold the farthest neighbors for each row index in the chunks;
Output:   none, an (intermediate) kNN graph stored in Gk';
row' ← blockIdx.x × blockDim.x + threadIdx.x;
if row' < n_chunksize then
    row ← split × n_chunkSize + row' // absolute row in the distance matrix;
    for column' ← 1 to n_chunksize do
        column ← chunk × n_chunkSize + column' //absolute column in the distance matrix;
        if row = column or row > n_row or column > n_col then
            continue /* exclude diagonal and pad regions */;
        if D[row' × n_chunksize + column'] < Gk'[Maxk[row']].weight then
            Gk'[Maxk[row']].source ← row;
            Gk'[Maxk[row']].target ← column;
            Gk'[Maxk[row']].weight ← D[row' × n_chunksize + column'];
            Search the new maximum element in row'(D) and store the index in Maxk[row'];
```

Figure 3.4.4: KNN pseudocode [46]

3.4.3 Support Vector Machine (SVM)

When there is a problem with n-dimensional input data X that is needed to be entered into a k-dimensional space Support Vector Machine (SVM) can be used to provide a non-linear mapping for the input. It will help in isolating the data by linear algebra and geometry theories. To find a classification for any point of data with a specified class, SVM can identify a hyperplane that will cover thins points.

$$y = \begin{cases} +1 \; if \; \overrightarrow{X}.\overrightarrow{\omega} + b \geq 1 \\ -1 \; if \; \overrightarrow{X}.\overrightarrow{\omega} + b < 1 \end{cases}$$

EQ 3.4.3: linear SVM class selection

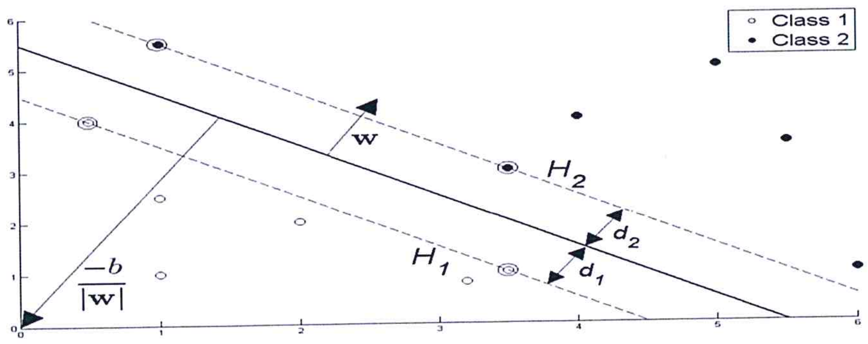

Figure 3.4.5: SVM

In the above figure Maximum-Margin Hyperplane and Margins for an SVM Trained with Samples from Two Classes [42]. Samples on the margin are called the Support Vectors. Following is an algorithm explaining how SVM works.

Data : Dataset with p^* variables and binary outcome.
Output: Ranked list of variables according to their relevance.

Find the optimal values for the tuning parameters of the SVM model;
Train the SVM model;
$p \leftarrow p^*$;
while $p \geq 2$ **do**
  $SVM_p \leftarrow$ SVM with the optimized tuning parameters for the p variables and observations in **Data**;
  $w_p \leftarrow$ calculate weight vector of the SVM_p $(w_{p1}, \ldots, w_{pp})$;
  $rank.criteria \leftarrow (w_{p1}^2, \ldots, w_{pp}^2)$;
  $min.rank.criteria \leftarrow$ variable with lowest value in $rank.criteria$ vector;
  Remove $min.rank.criteria$ from **Data**;
  $Rank_p \leftarrow min.rank.criteria$;
  p ← p - 1 ;
end
$Rank_1 \leftarrow$ variable in **Data** $\notin (Rank_2, \ldots, Rank_{p^*})$;
return $(Rank_1, \ldots, Rank_{p^*})$

Figure 3.4.6: SVM pseudocode [47]

3.4.4 Ensemble

Accuracy is one of the most effective and important factors when it comes to evaluating any algorithm, and the Ensemble algorithm is a new algorithm that aims to increase the level of accuracy. The main idea of the ensemble algorithm is to create a model that combines multiple machine learning models. The Ensemble algorithm has two categories: parallel ensemble, and sequential ensemble. The parallel technique relies on the idea of having multiple algorithms working independently and then aggregating the results and updating the system wights by averaging the results, which allows each algorithm to show its strength without affecting others. The second technique is sequential, where the modules are arranged in sequential order, each one feeding the next with its results, which will allow the algorithm to have better detection of the errors and the wights will be assigned in better ways. [48]

$$y = argmax_{C_j \in C} \sum_{h_j \in H} P(C_j|h_j)P(T|h_j)P(h_j)$$ EQ 3.4.4: Ensemble classification

where y is the expected class, C is the group of all classes, H is the hypothesis space, P is the probability, and the training data is referred to as T.

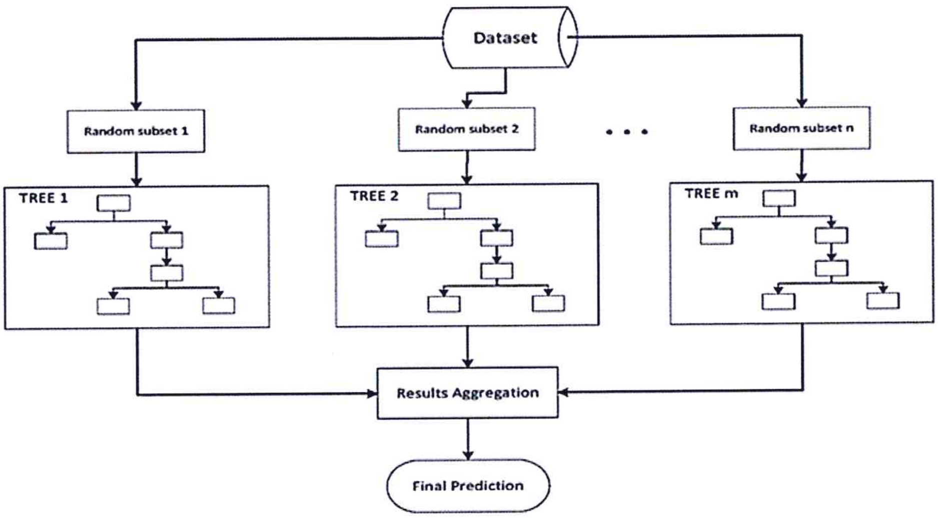

Figure 3.4.7: Ensemble Structure

There are different algorithms and implementations for ensemble, the following is general pseudocode explaining the algorithm.

Algorithm 1 Pseudocode of the ensemble learning algorithm

Input: Training dataset $\mathcal{D} = \{(x_1, c_1), (x_2, c_2), ..., (x_n, c_n)\}$
Base level classifiers $\mathcal{L}1, ..., \mathcal{L}_k$
Meta level classifier $\hat{\mathcal{L}}$
Output: Trained ensemble classifier $\hat{\mathcal{M}}$
BEGIN
Step 1: Train base learners by applying classifiers $\mathcal{L}_i$ to dataset $\mathcal{D}$
for $i = 1, ..., k$ **do**
$\mathcal{B}_i = \mathcal{L}_i(\mathcal{D})$
end for
Step 2: Construct new dataset of predictions $\hat{\mathcal{D}}$
for $j = 1, ..., n$ **do**
for $i = 1, ..., k$ **do**
% use $\mathcal{B}_i$ to classify training example x_j
$z_{ij} = \mathcal{B}_i(x_j)$
end for
$\hat{\mathcal{D}} = \{\mathcal{Z}_j, c_j\}$, where $\mathcal{Z}_j = \{z_{ij}, z_{2j}, ..., z_{nj}\}$
end for
Step 3: Train a meta level classifier $\hat{\mathcal{M}}$
$\hat{\mathcal{M}} = \hat{\mathcal{L}}(\hat{\mathcal{D}})$
Return $\hat{\mathcal{M}}$
END

Figure 3.4.8: Ensemble pseudocode [49]

3.4.5 Convolutional Neural Networks (CNN)

Deep Learning is a part of machine learning, its learning relies on understanding the representation level of data. One of the most popular types of deep learning is the Convolutional Neural Networks (CNN), which proved its power with the complex types of data, such as images, time series, speech signals, text, and many others. The early use of CNNs started in the early nineties [50][51] for character recognition, and rose from that time to solve many complex problems.

Multilayer perceptrons are usually referring to the fully connected neural network, where each neuron in a layer is connected to all neurons in the next layer. The high number of connections can lead these networks towards overfitting. To resolve this issue, CNNs used the regularization process, and that's why CNN's are considered a regularized version of MLPs. There are multiple methods to regularize the network, for example: dropping some connections to reduce the connectivity and handle the parameters of the network during the training process, like weights decay.

CNN usually requires much less preprocessing for the input data when compared to other classification methods. This means that CNNs can learn how to optimize the kernels (or filters), through unsupervised learning, while in other traditional algorithms, the filters and manually created and engineered to match the problem.

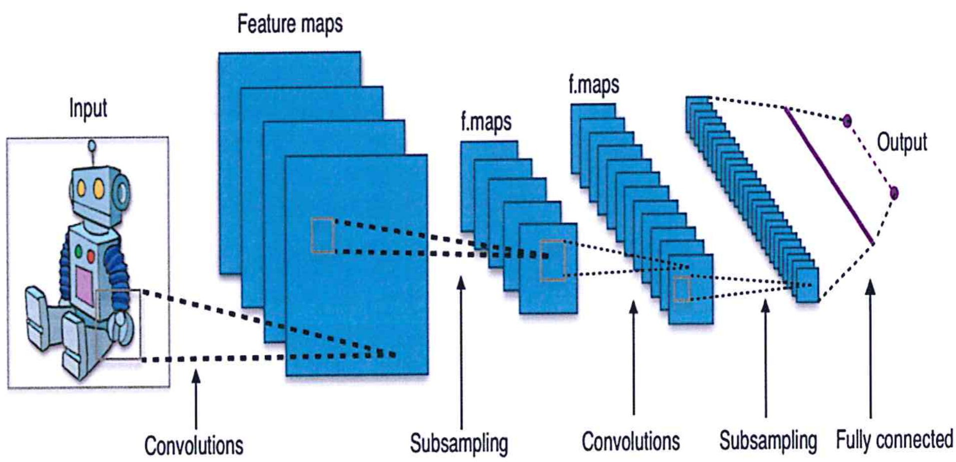

Figure 3.4.9: Typical CNN architecture

The typical structure of a convolutional neural network is the input layer, set of hidden layers, and output layer. A layer is considered to be a hidden layer in a feed-forward neural network if it locates between the input layer and the output layer. They are called hidden layers because the inputs and outputs are masked by the final convolution and the activation function.

There are several types of CNNs layers and follows are the main layers:

Convolutional layers: this layer takes the input as raw data, convolves it, and passes it to the next layer. After the data passes through the convolutional layers, it will be converted to a feature map (which can also be called an activation map).

Following are the main equations for this layer:

1- Computing the contributions from the previous layers cell in forwarding propagation

$$x_{ij}^{l} = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} \omega_{ab} y_{(i+a)(j+b)}^{l-1}$$ EQ 3.4.5: contribution of each neuron

2- Calculate the nonlinearity (y) in the convolution layer in forwarding propagation

$$y_{ij}^{l} = \sigma(x_{ij}^{l})$$ EQ 3.4.6: Applying CNN nonlinearity

3- Calculate the error (E) for each neuron in the backward propagation

$$\frac{\partial E}{\partial \omega_{ab}} = \sum_{i=0}^{N-m} \sum_{j=0}^{N-m} \frac{\partial E}{\partial x_{ij}^{l}} y_{(i+a)(j+b)}^{l-1}$$ EQ 3.4.7: Backward propagation error

4- To be able to calculate the gradient, it's important to know the value of deltas $(\frac{\partial E}{\partial x_{ij}^{l}})$

$$\frac{\partial E}{\partial x_{ij}^{l}} = \frac{\partial E}{\partial y_{ij}^{l}} \sigma'(x_{ij}^{l})$$ EQ 3.4.8: Deltas

5- To propagate the errors back, we can use the following equation

$$\frac{\partial E}{\partial y_{ij}^{l-1}} = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} \frac{\partial E}{\partial x_{(i-a)(j-b)}^{l}} \omega_{ab}$$ EQ 3.4.9: Propagate error

Pooling layers: The main aim here is to reduce the complexity of the data by reducing the dimensions of the data, where the output of a group or cluster of neurons in the previous layer is converted into an input for a single neuron in the next layer.

Fully connected layers: Most CNNs have a traditional multi-layer perceptron neural network.

3.5 CNN Structure

Deep learning is one of the most advanced methods of machine learning, and CNN is one important part of deep learning. It allows the network to convolute the input data to be able to do lots of processing steps easily, for example, the CNN allows the system to be able to detect the features and extract the important ones.

Also, it will allow the system to find patterns that reduce human interaction and it will be able to adapt to changes and new features.

There are many types of deep learning networks and even different types of layers, and each layer is used for a specific work, for example, the Conv1D layer is used when the input has one diminution, while Conv2D is used when there is more than one diminution, like images.

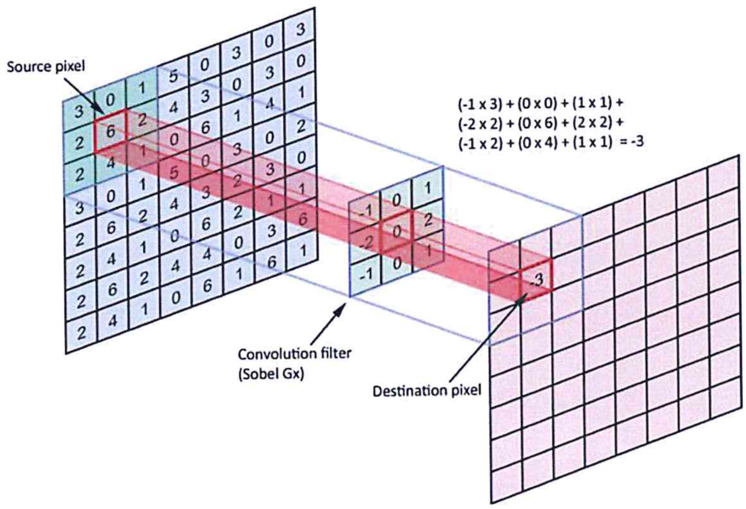

Figure 3.5.1: Convolution Layer

The LeakyReLU layer is different from the activation function that is used inside the layer, where this layer is used with an alpha value to prevent the system from removing important values by having a small sloop for negative values instead of a flat zero.

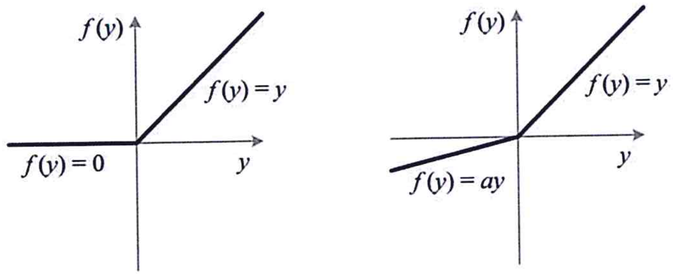

Figure 3.5.2: ReLU vs Leaky ReLU

Another important layer is the AveragePooling1D layer, which is used to reduce the number of input samples to make it easier to work with the data.

Also, the Dense layer is an important part, which is a fully-connected neural network. The CNN can also have an LSTM layer. The structure in figure 3.5.3 was used in this research.

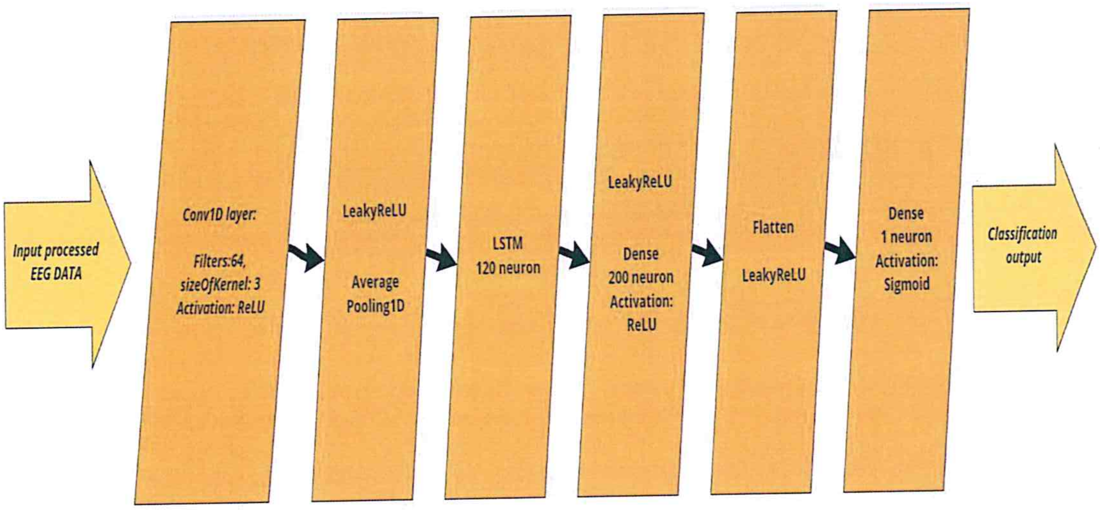

Figure 3.5.3: CNN structure

The convolutional neural network contains the following layers:

- Conv1D: input convolutional layer with 64 filters, a kernel size = 3 and activation ReLU activation function
- LeakyReLU: a layer used to give the output of the Conv1D layer a small shift to avoid

removing important points, with an alpha value = 0.2

- AveragePooling1D: a layer creates another windowing process to reduce the number of data points. In this case, the window size was set to 3
- LeakyReLU: a layer used to give the output of the AveragePooling1D layer a small shift to avoid creating unusable data points, with an alpha value = 0.2
- LSTM: Long short-term memory (LSTM) layer is used in deep learning modules usually because of the features it has like the ability to process a sequence of data points and the feedback connections.
- LeakyReLU: a layer used to give the output of the LSTM layer a small shift to remove unwanted points without removing the important points, with an alpha = 0.2
- Dense: a densely (fully) connected neural network layer with 200 neurons, and ReLU activation function.
- Flatten: a layer used to flatten the output and make it more consistent.
- LeakyReLU: a layer used to give the output of Flatten layer a small shift to avoid removing data points, with an alpha value = 0.2
- Dense: a layer with one neuron that will be used as the output layer with the activation Sigmoid.

The model was trained on each patient separately, with 70% training data and 30% for validation and testing. The information that was extracted from the model is the accuracy, true positive, true negative, false positive, and false negative.

To make sure the model is functioning well, each cycle of training, testing, and validation was done 10 times for each patient, and the results were averaged. This shows that the results of the system are consistent and real. Also, it can be used as a validation method.

3.6 Metrics Selection

There are many ways to measure the correctness or performance of "pattern recognition and classification" neural networks [52]. In our work we will focus on the following to test the proposed model performance: Accuracy, True positive (TP), True negative (TN), False positive (FP), False negative (FN), Sensitivity, and Specificity.

To make it easier to understand the semantics, next we will be using the following abbreviations:

TP: The number of epileptic signals correctly classified as epileptic.

FP: The number of normal signals incorrectly classified as epileptic.

TN: The number of normal signals is correctly classified as normal.

FN: The number of epileptic signals incorrectly classified as normal.

Accuracy: The percentage of how much the model is accurate in classifying signals as epileptic and not epileptic. It's represented by the following equation:

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN}$$ EQ 3.6.1: Accuracy

Sensitivity (Recall): The percentage of signals that are classified correctly as epileptic to all signals that are classified as epileptic. It is calculated with the following equation:

$$Recall = \frac{TP}{TP+FN}$$ EQ 3.6.2: Sensitivity

Specificity: The percentage of signals correctly classified as normal to all signals classified as normal. Following its equation:

$$Specificity = \frac{TN}{TN+FP}$$ EQ 3.6.3: Specificity

Precision: The percentage of signals correctly classified as epileptic to all epileptic signals predicted.

$$Precision = \frac{TP}{TP+FP}$$ EQ 3.6.4: Precision

G-mean [53]: It's used to measure the balance between classifications using the following formula:

$$G - mean = \sqrt{Sensitivity * specificity}$$ EQ 3.6.5: G-mean

F-measuring [53]: it detects the level of balance between precision and sensitivity. It's calculated by the following equation:

$$F - measuring = \frac{2*sensitivity*precision}{senstivity+precision}$$ EQ 3.6.6: F-measuring

Chapter 4

Experiments and Results

4.1 Experiments and Results

After all previous steps, the proposed model was built and trained on the data, and in this section, we will present the results we got, compare them to other algorithms, and apply the metrics to evaluate the results.

To make sure that the models were trained correctly the following steps were taken:

1- The classification of regular algorithms was done through MATLAB classification learner with 7 cross-validations. Each algorithm has sub algorithms, so in each case, the highest value was taken

2- To test the proposed preprocessing algorithm and feature extraction effectivity, the CNN was tested two times once with the raw data after filtration and smoothing, and the other with the full algorithm.

3- To make sure that the results are correct, for each case the CNN was trained and tested 10 times, and the results were averaged.

4- All the work is done using HB Z-book workstation with *Intel(R) Core(TM) i7-4800MQ CPU @ 2.70GHz*, and 16GB ram.

4.2 CNN With Raw Data

One of the most important features of deep learning is its ability to extract the features from raw data and find the main actors that can differentiate between the types of output classes. In this experiment, the data was entered into the CNN as raw data of 22 inputs and one binary output. The processing that was done to this data is filtration, smoothing, and windowing, without extracting the frequency bands. The following table shows that the CNN was able to detect the features and give relatively good results. However, they are not good enough, and in the coming sections, we will show that some regular algorithms provided better accuracies. The following table shows the accuracy, True positive, True negative, False Positive, and False-negative for the 10 patients.

Patient	Accuracy	True Positive	True Negative	False Positives	False Negative	Sensitivity	Specificity
chb01	94.5	523	446	87	46	91.9	83.6
chb02	96.8	103	109	18	7	93.6	85.8
chb03	95.8	433	550	51	55	88.7	91.5
chb04	78.1	300	422	103	160	65.2	80.4
chb05	95.9	547	711	12	150	78.5	98.3
chb06	62.4	109	120	109	70	60.9	52.4
chb07	95.7	373	329	75	25	93.7	81.4
chb08	81.3	800	881	271	323	71.2	76.4
chb09	95.8	312	296	55	17	94.8	84.3
chb10	94.3	426	621	26	108	79.8	95.9

Table 4.2-1: CNN results with raw data

The results show that the system was able to handle the data, and was able to give relatively accepted accuracy with a total average accuracy of 89.03% for all the patients.

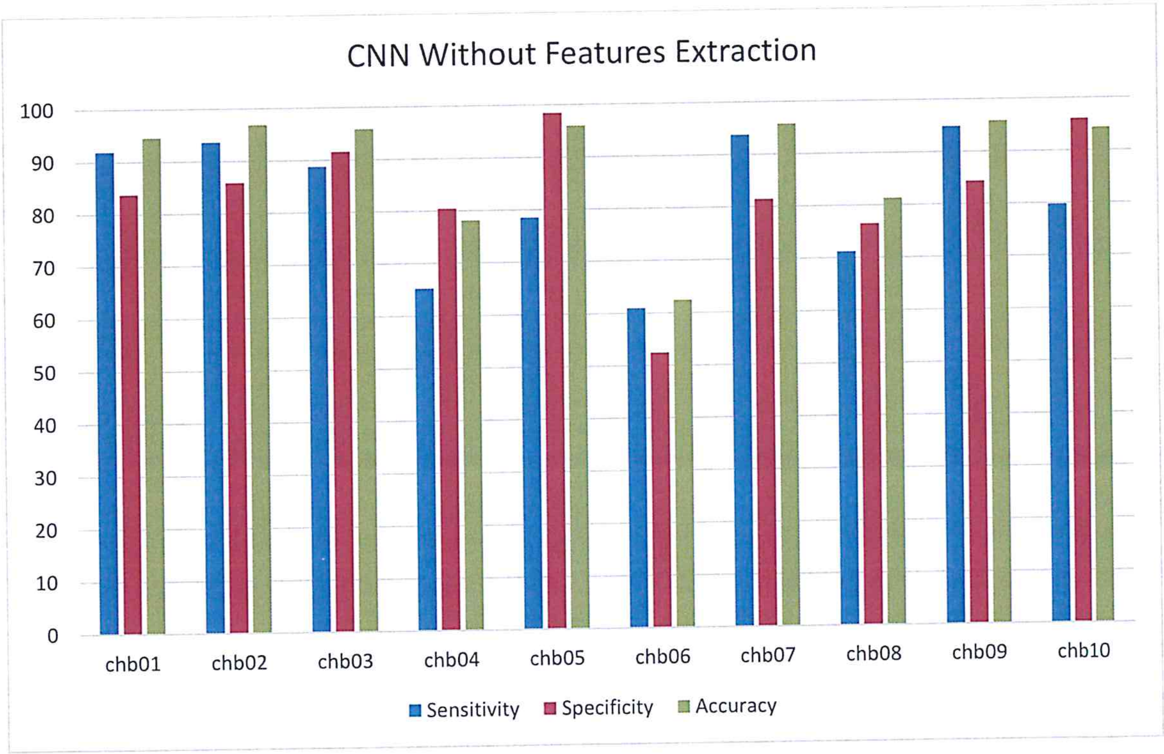

Figure 4.2.1: CNN Without Features Extraction results

From figure 4.2.1, it's clear that there is no consistency in the results, where the values alternate between 96.8 for CHB02 to 62.4% for CHB06. The values of sensitivity and specificity are consistent with the accuracy which indicates that the system was not able to handle the data of CHB06.

4.3 CNN With Frequency Bands Extracted

Running the CNN with the raw data shows that the data is complex and the CNN is not able to extract all the features that allow giving high accuracy without additional data handling. The step that was taken is the Frequency Bands Extraction (FBE).

Frequency Bands Extraction (FBE): is the process of converting the data from the time domain into the frequency domain, extracting the levels of frequency based on multiple factors like several inputs and signal frequency. Usually, the main five bands extracted are Delta, Theta, Alpha, Beta, and Gamma. For each row of the 22 inputs there will be five frequency values, which means that the data will be:

$$Data(points) = number\ of points\ in\ the\ record * 22\ inputs * 5\ frequency\ bands$$

So, the process of FBE increases the complexity of the data instead of reducing it. To overcome this issue the data were averaged based on the frequency; for each frequency, all the 22 inputs were averaged to provide one value, and that makes the data much smaller and simpler to handle:

$$Data(points) = number\ of points\ in\ the\ record * 5\ averaged\ frequency\ bands$$

As an example, for the case CHB01, the first record originally has 921,601 points and 22 data streams producing 20,275,222 points, which is a very small part of the data for a single patent. After the preprocessing and windowing of the data, it was reduced to 28,800 points and 22 data streams, producing 633,600 points. Although the process reduces the data points about 32 times; however, it was not enough, the data still large and the accuracy is not very high.

After FBE number of points was 28,800 points and 5 data streams, producing 144,000 points, which is about 4.4 times less than the processed data.

The data processing and FBE made the data easier to handle and more valuable because in the abnormal signals – epileptic signals in this case – the indicators can be detected easier when the data is in the frequency domain and the bands extracted, and this was proved by the results we have. The following table shows the results for the same data used in section 4.1 but after FBE.

Patient	Accuracy	True Positive	True Negative	False Positives	False Negative	Sensitivity	Specificity	Precision	G-mean	F-measuring
chb01	97.5	703	764	16	28	96.2	97.9	97.7	97	96.9
chb02	98.2	182	247	4	9	95.3	98.4	97.8	96.8	96.5
chb03	98.3	555	682	19	12	97.9	97.3	96.9	97.6	97.3
chb04	95.4	501	532	22	50	90.9	96	95.8	93.4	93.2
chb05	99.2	723	823	6	21	97.2	99.3	99.2	98.2	98.2
chb06	92	231	190	42	8	96.6	81.9	84.6	88.9	90.2
chb07	98.8	421	458	16	9	97.9	96.6	96.3	97.2	97.1
chb08	94.8	1011	1420	73	103	90.7	95.1	93.3	92.9	92
chb09	98.8	400	421	3	16	96.2	99.3	99.3	97.7	97.7
chb10	98.4	549	602	11	24	95.8	98.2	98	96.9	96.9

Table 4.3-1: CNN results with a features extraction algorithm

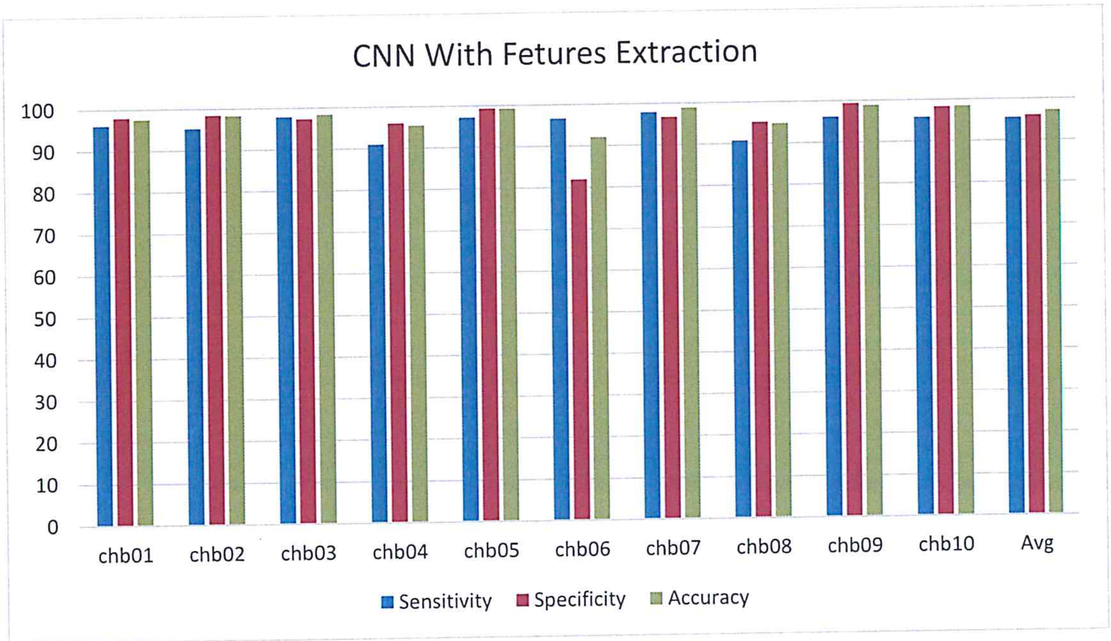

Figure 4.3.1: All patients' results for CNN with feature extraction algorithm

The system was able to give much better results after FBE, and it shows an average accuracy of 97.14%. From table 4.3-1 the accuracy level varies from 92% to 99.2%, which means that not all patents have the same results. Figure 4.3.1 also shows that the sensitivity and specificity levels are very class to the accuracy level, which leads us to the idea that the system is stable and can detect both normal and epileptic seizures.

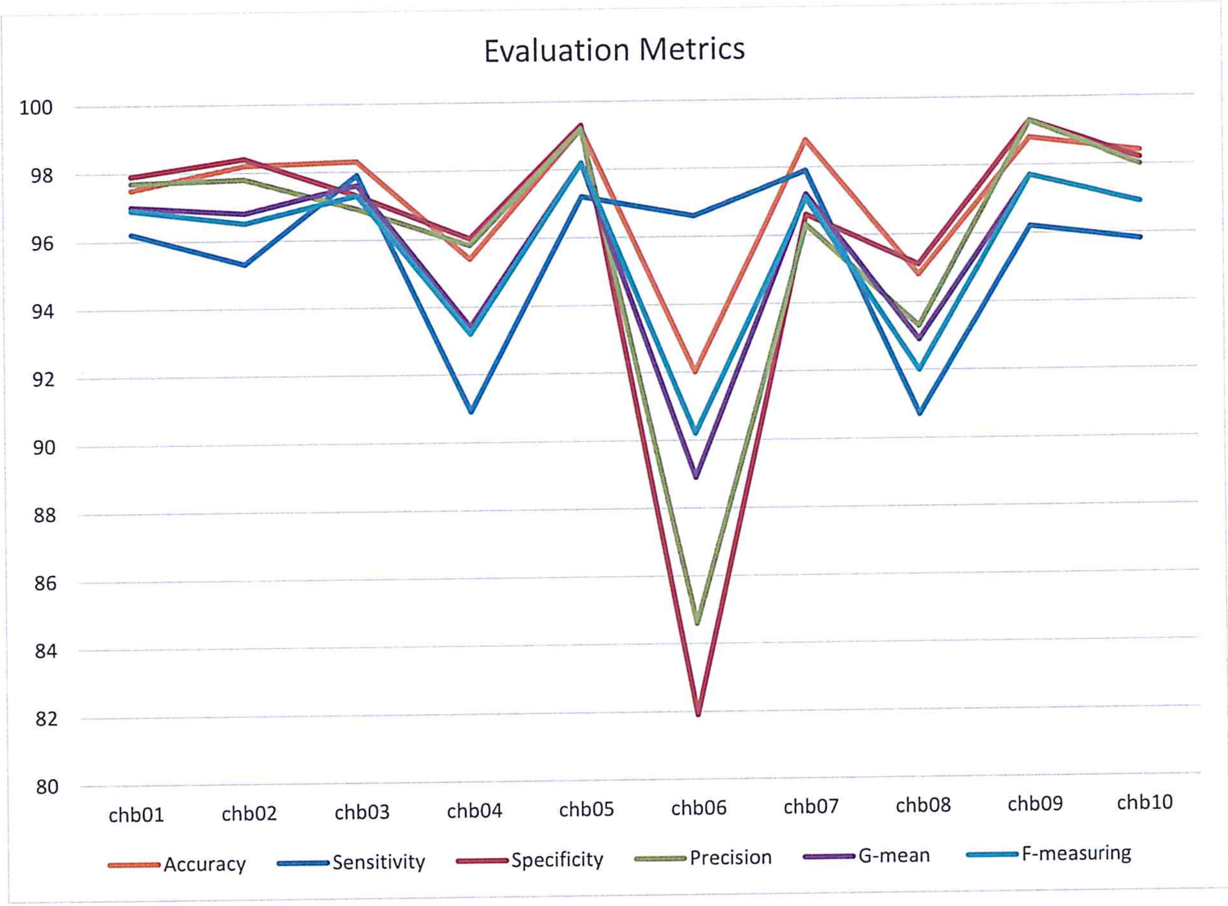

Figure 4.3.2: Evaluation Metrics

The data shown in figure 4.3.2 shows the metrics that were selected to evaluate the system. It's noticeable that the data is high in some cases and low in others. We can see that for CHB01, CHB02, and CHB03, the metrics lines are very close to each other with high values. However, in the case of CHB04, the sensitivity value drop to reach 90.9% while the specificity value reached 96%; this means that for this patient, the model was able to detect the normal singles with an accuracy of 96%, while it was able to detect the epileptic signals with an accuracy of 90.9%. From this, we can see that the value of accuracy is not the only factor in determining the quality of the system.

4.4 CNN VS Other Machine Learning Algorithms

In this part, the most popular classification algorithms were selected and used to classify the data produced by the preprocessing algorithm and the FBE. The following table shows the accuracy of each one of them. Each algorithm has its sub algorithms, to evaluate the system, the highest value was taken. The classification learner in MATLAB was used to do this process with 7 cross-validations.

Case	Tree	SVM	KNN	Ensemble	CNN With Raw Data	CNN With Bands
CHB01	92.3	91.9	91.7	92.8	94.5	97.5
CHB02	94.1	94.3	95.2	94.8	96.8	98.2
CHB03	94.5	94.8	94.9	94.9	95.8	98.3
CHB04	88.1	87.4	88.7	89.8	78	95.4
CHB05	95.9	95.9	96.3	96.5	95.9	99.2
CHB06	77.4	77	78.2	79.4	62.3	92
CHB07	95.3	95.5	96.6	96.2	95.7	98.8
CHB08	86.7	86.2	86.6	87.6	81.2	94.8
CHB09	96.5	95.4	96.9	97.1	95.8	98.8
CHB10	93.2	94	93	95.7	94.3	98.4
Average	91.4	91.24	91.81	92.48	89.03	97.14

Table 4.3-1: comparison between CNN and other algorithms

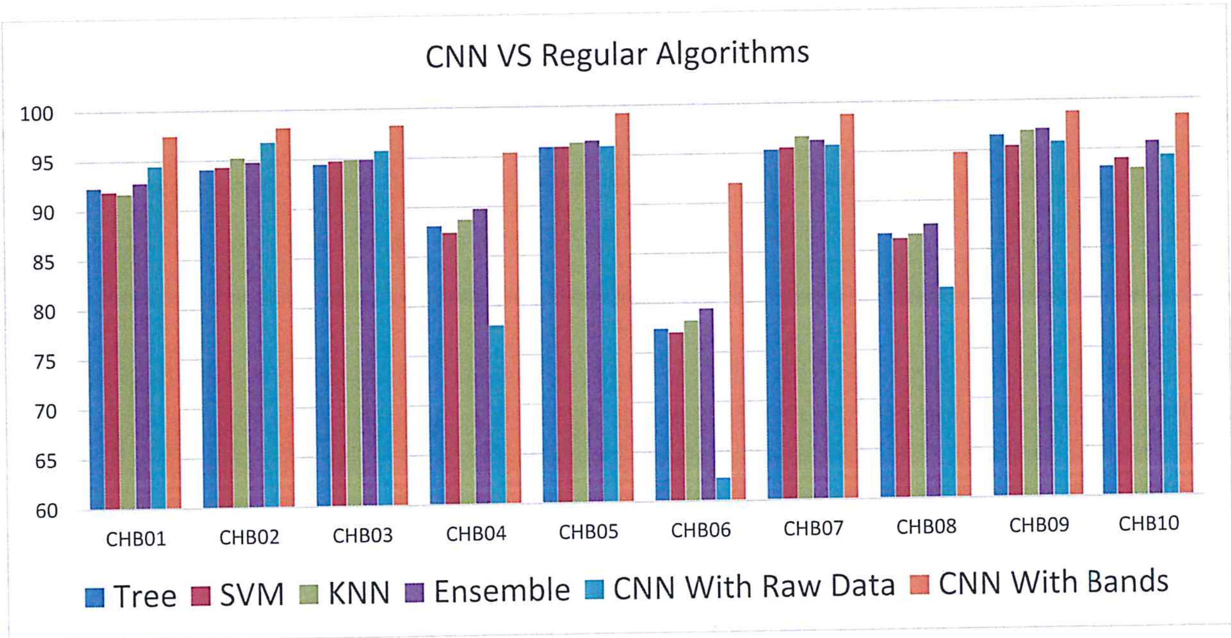

Figure 4.3.1: results for each patient in all algorithms

From figure 4.4.1, we can see a pattern that repeats in the data, where all the algorithms are giving their highest accuracy at the same patient, and lowest accuracy in the same way. This makes it very clear that the proposed patient-level approach is correct, and this can enhance the quality; where instead of mixing the information from CHB06 (lowest accuracy) with others and reducing the total accuracy, each case will maintain its accuracy. It's worth mentioning that, even though the CHB06 case has the lowest accuracy of all the algorithms, the proposed approach manages to reach 92% of accuracy, 96.6% sensitivity, and 81% specificity, which means that in the worst case, the model was able to diagnose the epileptic signals with accuracy if 96.6%, with an error rate of 19%.

4.5 Discussion

From the previous sections very different results were obtained, to be able to understand the results, they must be studied in two-way, Algorithm level, and Case level. At the Algorithm level, the worst one was the CNN with Raw Data with an average accuracy of 89.03%, and that can be explained due to the huge amount of input data since it was used as it is without the frequencies extracted, which added more complexity to the data and make the process of classification very hard. This showed that the process of using the raw input data is not efficient, and it needs either to change the system structure or to handle the data in a way that gives a better result; data handling was chosen in this work.

Tree Algorithms, KNN, and Ensemble gave better results with an average accuracy of 91.4%, 91.81%, and 92.48% respectively. This can be explained in general by the main idea of these algorithms, which is having multiple reference points that can help in giving better results.

Although the results were better, the accuracy is still low even though the data used in these algorithms is the frequency processed data. The best results obtained in this study were the results from the CNN with Bands Algorithm. The proposed algorithm was able to give the best accuracy with an average of 97.14% which is the best result among all the algorithms used in this study. The size of the data, in this case, was much less than the one in the CNN with Raw Data Algorithm, even though the CNN require more data usually to train better and give more accurate results, however, in this case, the smaller data contained richest information, and the CNN with Bands Algorithm was able to give many accurate results in less processing time, and less amount of data.

When it comes to the accuracy in the above discussion, the average was used and that's because the main idea of the proposed algorithm is to be trained and used tested on each case independently. The justification of this process was shown in the results and will be presented following. Since EEG data tests generate a huge amount of data, it will be easy to gather input data for the same patient to allow the AI algorithms to be fully trained. Also, for each patient, the same activities or signals can vary in length, amplitude, and sharpness, which makes it very difficult to find a general format or rule to classify signals. And these are the main reasons to train the algorithms at the Case Level.

One of the main noticeable things in table 4.4-1 is that for the same case there is some consistency in the results. For example, in Case CHB06, all the algorithms gave the lowest accuracy, and for case CHB09 the accuracy was the highest at most of the algorithms. These patterns are not coincidences, and after representing the data, it showed as follows:

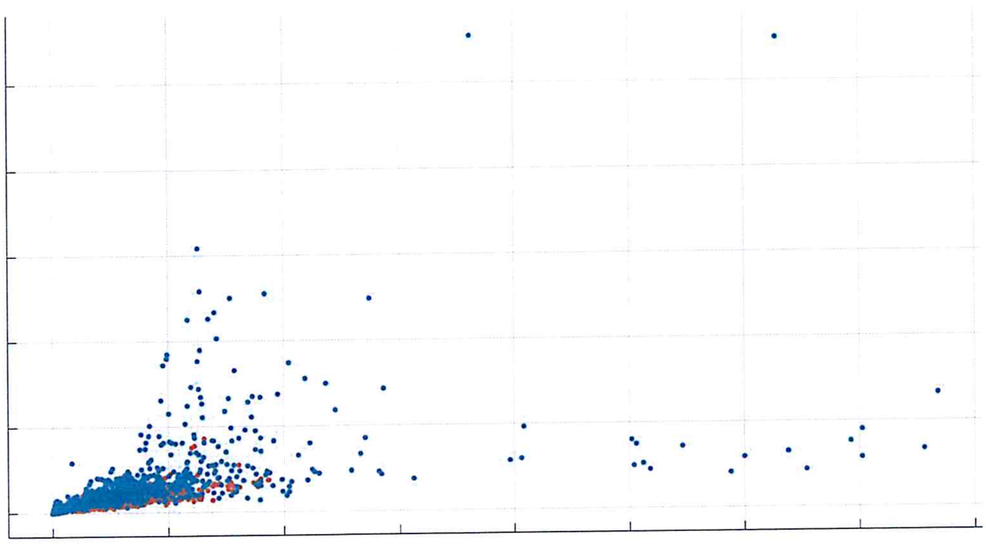

Figure 4.3.1: CHB06 records points are very close to each other

From the above figure, it's clear that the data points are very close to each other, and the classification is very hard, which causes the classification accuracy to be low in all the used algorithms.

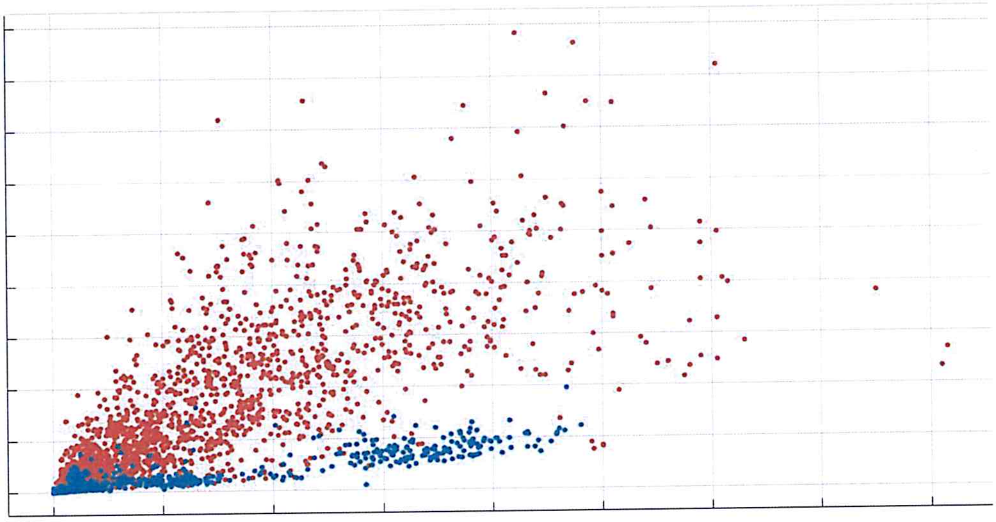

Figure 4.3.2: CHB09 records points are very far from each other

In the CHB09 case which is shown the above figure, the data points distribution is better and it's easier to classify such cases. And that can explain why all the algorism got high accuracy for this case.

Theoretically, each person has a unique brain, and its activities are not the same, even the abnormal and the ones that are related to diseases can show in different ways. And if the system was trained in mixed cases, it could not be the best way to get a real-world system. Although it will be more general, it will be less accurate

Chapter 5

Conclusion and Future work

5.1 Conclusion and Future Works

The human brain is the most complex part of the human body, and it's the control unit for all the other systems of the body. Any problem in the controlling unit can cause serious problems in the other systems, and that's one of the most important things that drive the researchers to target the human brain in their studies. One of the most important problems related to the brain is epilepsy; Which is an abnormal activity in the brain that leads to seizures and can cause involuntary movements. Epilepsy is studied heavily because it can result in serious problems in the patient's life due to the seizures and the fact that it can happen at any time. Also, the epileptic patterns are unique and can be detected easier than other neurological problems.

To study the brain an Electroencephalogram (EEG) recording is done to record the electrical activities inside. These records are studied by doctors and neurologists to evaluate the condition of the brain and also to diagnose if there are problems in it such as epilepsy.

Machine learning is one of the Advanced artificial intelligences (AI) algorithms that aim to give computers or machines the ability to mimic human thinking actions and to be able to learn. This process will allow the machine to consume data, analyze it, and make a design based on it. This feature will help humans in solving complex problems and give human-like suggestions based on historical data and complex mathematical calculations.

One of the most advanced machine learning algorithms is deep learning, which is an algorithm that can detect the features in the data without providing it to the learning algorithm. This algorithm is used with big data because it needs large sets of data to be able to obtain the features on its own.

Detecting the epileptic waves in an EEG record is not an easy job, because usually, the record contains 22 data rows where each second is sampled at a large rate of points per second – usually 256 – and sometimes the length of a single record can reach more than one hour. All of that makes it very hard to detect abnormal signals in the EEG record.

To overcome these issues, computer programs are used to help in the diagnosis process and to do so, intelligent algorithms are needed.

From the above, it's convenient to use CNN with EEG data to help classify and diagnose epileptic cases, and that's what has been done in this work.

A simplified CNN algorithm was proposed to classify the EEG signals as epileptic or nonepileptic signals. The algorithm contains all the processes from reading the raw data to doing the pre-processing in the data, classifying it, and giving the results.

The main idea of the work in EEG with this algorithm is to customize it based on each patient case so that the algorithm will be tuned to detect the abnormality for the case itself based on the patient's historical data. This process aims to have a more accurate algorithm than the general algorithms.

The testing of the proposed work was done on 10 cases, where each case has multiple records with both epileptic and nonepileptic signals. The testing was done two times, one with the data after it's being processed and filtered, and the other one with the data filtered without processing.

The algorithm was able to achieve an average accuracy of (97.14%) with the processed data, and (89.03) with no processed data. The results were compared to other algorithms such as Tree (91.4%), SVM (91.24%), KNN (91.81), and Ensemble (92.48%). And that shows that the results of the proposed algorithm are much better than the others. Also, when comparing the research results with other researches which used the same dataset with a similar methodology like research [37] where they manage to get a total classification accuracy of 96.74%, It's clear that our research is giving high accuracy with less complicated structure.

The main reason for making the algorithm work at the case level is to be able to tune it based on the patient brain activities. The results show that in the same algorithm with almost the same amount of data for two different cases the accuracy of classification reached 99.2% for CHB05, then dropped to 92% for CHB06 due to the differences in the structure of the data for each case.

5.2 Future Work

We present in this thesis a model for detecting epileptic signals in the EEG records. This can be a good base to do more research in this field, and enhance the research model to be applied in real-life applications. In the future, we will continue in multiple directions with our research, and some of them are the following:

1- Enhance the model accuracy

2- Enhance the model to be able to detect several abnormal brain signals.

3- Gather local EEG records and try to create a national dataset that can be used to understand and analyze the local neurological problems.

4- Create a prediction model that can predict the seizures before they occur.

And many other enhancements and comparisons can be done, like changing the preprocessing methods, comparing our work to similar models using the same dataset, etc.

6 Bibliography

[1] Wang, S. C. (2003). Artificial neural network. In Interdisciplinary computing in java programming (pp. 81-100). Springer, Boston, MA.

[2] Nykopp, T. (2001). Statistical modelling issues for the adaptive brain interface. Helsinki: Helsinki University of Technology.

[3] Azevedo, F. A., Carvalho, L. R., Grinberg, L. T., Farfel, J. M., Ferretti, R. E., Leite, R. E., ... & Herculano-Houzel, S. (2009). Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. Journal of Comparative Neurology, 513(5), 532-541.

[4] Anderson, C. W., & Sijercic, Z. (1996, June). Classification of EEG signals from four subjects during five mental tasks. In Solving engineering problems with neural networks: proceedings of the conference on engineering applications in neural networks (EANN'96) (pp. 407-414). Turkey.

[5] Berka, C., Levendowski, D. J., Cvetinovic, M. M., Petrovic, M. M., Davis, G., Lumicao, M. N., ... & Olmstead, R. (2004). Real-time analysis of EEG indexes of alertness, cognition, and memory acquired with a wireless EEG headset. International Journal of Human-Computer Interaction, 17(2), 151-170.

[6] Chaovalitwongse, W. A., Pottenger, R. S., Wang, S., Fan, Y. J., & Iasemidis, L. D. (2011). Pattern- and network-based classification techniques for multichannel medical data signals to improve brain diagnosis. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 41(5), 977-988.

[7] https://www.cham.org/specialties-and-programs/neurology/conditions/comprehensive-epilepsy-center

[8] https://www.epilepsy.com/learn/about-epilepsy-basics/what-epilepsy

[9] Chaovalitwongse, W. A., Pottenger, R. S., Wang, S., Fan, Y. J., & Iasemidis, L. D. (2011). Pattern- and network-based classification techniques for multichannel medical data signals to improve brain diagnosis. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 41(5), 977-988.

[10] Gajic, D., Djurovic, Z., Di Gennaro, S., & Gustafsson, F. (2014). Classification of EEG signals for detection of epileptic seizures based on wavelets and statistical pattern recognition. Biomedical

Engineering: Applications, Basis and Communications, 26(02), 1450021.

[11] Schirrmeister, R. T., Springenberg, J. T., Fiederer, L. D. J., Glasstetter, M., Eggensperger, K., Tangermann, M., ... & Ball, T. (2017). Deep learning with convolutional neural networks for EEG decoding and visualization. Human brain mapping, 38(11), 5391-5420.

[12] Fisher, R. S., Acevedo, C., Arzimanoglou, A., Bogacz, A., Cross, J. H., Elger, C. E., ... & Wiebe, S. (2014). ILAE official report: a practical clinical definition of epilepsy. Epilepsia, 55(4), 475-482.

[13] Kakisaka, Y., Alexopoulos, A. V., Gupta, A., Wang, Z. I., Mosher, J. C., Iwasaki, M., & Burgess, R. C. (2011). Generalized 3-Hz spike-and-wave complexes emanating from focal epileptic activity in pediatric patients. Epilepsy & Behavior, 20(1), 103-106.

[14] Nardello, R., Plicato, G., Mangano, G. D., Gennaro, E., Mangano, S., Brighina, F., ... & Fontana, A. (2020). Two distinct phenotypes, hemiplegic migraine, and episodic Ataxia type 2, caused by a novel common CACNA1A variant. BMC neurology, 20(1), 1-7.

[15] Halford, J. J., Schalkoff, R. J., Zhou, J., Benbadis, S. R., Tatum, W. O., Turner, R. P., ... & Dean, B. C. (2013). Standardized database development for EEG epileptiform transient detection: EEGnet scoring system and machine learning analysis. Journal of neuroscience methods, 212(2), 308-316.

[16] Benbadis, S. R. (2007). Errors in EEGs and the misdiagnosis of epilepsy: importance, causes, consequences, and proposed remedies. Epilepsy & Behavior, 11(3), 257-262.

[17] Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P. C., Mark, R., ... & Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation [Online]. 101 (23), pp. e215–e220.

[18] Shoeb, A. H. (2009). Application of machine learning to epileptic seizure onset detection and treatment (Doctoral dissertation, Massachusetts Institute of Technology).

[19] Rojas, G. M., Alvarez, C., Montoya, C. E., de la Iglesia-Vayá, M., Cisternas, J. E., & Gálvez, M. (2018). Study of resting-state functional connectivity networks using EEG electrodes position as seed. Frontiers in neuroscience, 12, 235.

[20] Sri, K. S., & Rajapakse, J. C. (2008, June). Extracting eeg rhythms using ica-r. In 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational

Intelligence) (pp. 2133-2138). IEEE.

[21] Gotman, J. (1982). Automatic recognition of epileptic seizures in the EEG. Electroencephalography and clinical Neurophysiology, 54(5), 530-540.

[22] Qu, H., & Gotman, J. (1997). A patient-specific algorithm for the detection of seizure onset in long-term EEG monitoring: possible use as a warning device. IEEE transactions on biomedical engineering, 44(2), 115-122.

[23] Adeli, H., Ghosh-Dastidar, S., & Dadmehr, N. (2007). A wavelet-chaos methodology for analysis of EEGs and EEG subbands to detect seizure and epilepsy. IEEE Transactions on Biomedical Engineering, 54(2), 205-211.

[24] Güler, N. F., Übeyli, E. D., & Güler, I. (2005). Recurrent neural networks employing Lyapunov exponents for EEG signals classification. Expert systems with applications, 29(3), 506-514.

[25] Übeyli, E. D. (2006). Analysis of EEG signals using Lyapunov exponents. Neural Network World, 16(3), 257

[26] Tzallas, A. T., Tsipouras, M. G., & Fotiadis, D. I. (2007). Automatic seizure detection based on time-frequency analysis and artificial neural networks. Computational Intelligence and Neuroscience, 2007.

[27] Ghosh-Dastidar, S., Adeli, H., & Dadmehr, N. (2008). Principal component analysis-enhanced cosine radial basis function neural network for robust epilepsy and seizure detection. IEEE Transactions on Biomedical Engineering, 55(2), 512-518.

[28] Zhou, J., Schalkoff, R. J., Dean, B. C., & Halford, J. J. (2012, August). Morphology-based wavelet features and multiple mother wavelet strategy for spike classification in EEG signals. In 2012 Annual international conference of the IEEE engineering in medicine and biology society (pp. 3959-3962). IEEE.

[29] Rajendran, S. (2016). Identification and use of psd-derived features for the contextual detection and classification of eeg epileptiform transients (Doctoral dissertation, Clemson University).

[30] Bagalore, K. S. (2016). Studying the use of hidden markov models in the detection and classification of eeg epileptiform transients using lpc features (Doctoral dissertation, Clemson University).

[31] Jagadeesha, R. (2017). Detection and classification of eeg epileptiform transients with rbf networks using hilbert huang transform-derived features (Doctoral dissertation, Clemson University).

[32] Yıldırım, Ö., Baloglu, U. B., & Acharya, U. R. (2018). A deep convolutional neural network model for automated identification of abnormal EEG signals. Neural Computing and Applications, 1-12.

[33] Saidi, A., Othman, S. B., Kacem, W., & Saoud, S. B. (2018, March). FPGA Implementation of EEG Signal Analysis System for the Detection of epileptic seizure. In 2018 International Conference on Advanced Systems and Electric Technologies (IC_ASET) (pp. 415-420). IEEE.

[34] Shoeb, A., Edwards, H., Connolly, J., Bourgeois, B., Treves, S. T., & Guttag, J. (2004). Patient-specific seizure onset detection. Epilepsy & Behavior, 5(4), 483-498.

[35] Acharya, U. R., Oh, S. L., Hagiwara, Y., Tan, J. H., & Adeli, H. (2018). Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals. Computers in biology and medicine, 100, 270-278.

[36] Andrzejak, R. G., Lehnertz, K., Mormann, F., Rieke, C., David, P., & Elger, C. E. (2001). Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. Physical Review E, 64(6), 061907.

[37] Kaziha, O., & Bonny, T. (2020). A Convolutional Neural Network for Seizure Detection. In 2020 Advances in Science and Engineering Technology International Conferences (ASET) (pp. 1-5). IEEE.

[38] Addison, P. S. (2005). Wavelet transforms and the ECG: a review. Physiological measurement, 26(5), R155.

[39] Chen, G., & Hou, R. (2007, August). A new machine double-layer learning method and its application in non-linear time series forecasting. In 2007 International Conference on Mechatronics and Automation (pp. 795-799). IEEE.

[40] Breiman, L., Friedman, J., Olshen, R. A., & Stone, C. J. (1984). Classification and regression trees Chapman & Hall. New York.

[41] Speybroeck, N. (2012). Classification and regression trees. International journal of public

health, 57(1), 243-246.

[42] Huang, C., & Townshend, J. R. G. (2003). A stepwise regression tree for nonlinear approximation: applications to estimating subpixel land cover. International Journal of Remote Sensing, 24(1), 75-90.

[43] Parvin, H., Alizadeh, H., & Minaei-Bidgoli, B. (2008, October). MKNN: Modified k-nearest neighbor. In Proceedings of the world congress on engineering and computer science (Vol. 1). Newswood Limited.

[44] Dasarathy, B. V. (1991). Nearest neighbor (NN) norms: NN pattern classification techniques. IEEE Computer Society Tutorial.

[45] Wu, Y., Ianakiev, K., & Govindaraju, V. (2002). Improved k-nearest neighbor classification. Pattern recognition, 35(10), 2311-2318.

[46] Arefin, A. S., Riveros, C., Berretta, R., & Moscato, P. (2012). Gpu-fs-k nn: A software tool for fast and scalable k nn computation using gpus.

[47] Sanz, H., Valim, C., Vegas, E., Oller, J. M., & Reverter, F. (2018). SVM-RFE: selection and visualization of the most relevant features through non-linear kernels. BMC bioinformatics, 19(1), 1-18.

[48] Rokach, L. (2010). Ensemble-based classifiers. Artificial intelligence review, 33(1), 1-39.

[49] Azeez, N. A., Odufuwa, O. E., Misra, S., Oluranti, J., & Damaševičius, R. (2021, March). Windows PE Malware Detection Using Ensemble Learning. In Informatics (Vol. 8, No. 1, p. 10). Multidisciplinary Digital Publishing Institute.

[50] Le Cun, Y., Bottou, L., & Bengio, Y. (1997, April). Reading checks with multilayer graph transformer networks. In 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (Vol. 1, pp. 151-154). IEEE.

[51] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[52] Nguyen, G. H., Bouzerdoum, A., & Phung, S. L. (2009). Learning pattern classification tasks with imbalanced data sets. Pattern recognition, 193-208.

[53] Akosa, J. (2017, April). Predictive accuracy: A misleading performance measure for highly

imbalanced data. In Proceedings of the SAS Global Forum (Vol. 12).

7 Appendix

The appendix includes the result of optimization experiments that performed on Z-Alizadeh Sani dataset.

7.1 Data preprocessing without features extraction code

```
from numpy.lib.function_base import average
import pandas as pd
import mne
import numpy as np
import matplotlib.pyplot as plt
from biosppy.signals.eeg import eeg
from biosppy.signals.tools import windower
from biosppy.signals import tools as st
import collections
import os

def _calc_vals(vals):
    c = collections.Counter(vals)
    if c[0.0] >= len(vals)/2:
        return 0
    return 1
def _avg(vals):
    res = vals[0]
    for row in vals:
        for i in range(0, len(row)):
            res[i] = (res[i] + row[i])/2.0
    return res
faf = 0
for oo in range(1, 11):
    n = str(oo)
    n = n.zfill(2)
    print("Working With chb", n)
    if not os.path.exists("results_no_band/chb%s" % n):
        os.mkdir("results_no_band/chb%s" % n)
    d = pd.read_csv("results/chb%s_res.csv" % n, index_col=0)
    for col in d.columns:
        print("Working With chb", n, col)

        fileName = "path_to_the_data_file/%s_%s" % (n, col.strip())
        data2 = mne.io.read_raw_edf(fileName)
        data1 = data2.get_data()
        data1 = np.array(data1).T

        data1_res = np.array(d[col])
        sampling_rate = 256
        data1_res = [x for x in data1_res if str(x) != "nan"]
        length = len(data1_res)
        overlap = 0.5
        wind_size = 0.25
        wind_size = int(wind_size * sampling_rate)
        step = wind_size - int(overlap * wind_size)
        nb = 1 + (length - wind_size) // step
        r = []
        index, values = windower(signal=data1_res,
                                 size=wind_size,
                                 step=step,
                                 kernel='hann',
                                 fcn=_calc_vals,)
```

```
data1 = np.array(data1)[:, 0:23]

# high pass filter
b, a = st.get_filter(ftype='butter',
                band='highpass',
                order=8,
                frequency=4,
                sampling_rate=sampling_rate)
aux, _ = st._filter_signal(
    b, a, signal=data1, check_phase=True, axis=0)

# low pass filter
b, a = st.get_filter(ftype='butter',
                band='lowpass',
                order=16,
                frequency=40,
                sampling_rate=sampling_rate)

filtered, _ = st._filter_signal(
    b, a, signal=aux, check_phase=True, axis=0)

_, data1_processed = windower(signal=filtered,
                    size=wind_size,
                    step=step,
                    kernel='hann',
                    fcn=_avg,)
final_vals = {}
for k in range(1, 24):
    final_vals[k] = []
for k in range(0, len(data1_processed)):
    for i in range(0, 23):
        final_vals[i+1].append(data1_processed[k][i])

final_vals['result'] = values
df = pd.DataFrame(final_vals)
filepath = 'results_no_band/chb%s/%s_bans_excel_file_test.csv' % (
    n, col.strip().split('.')[0])
df.to_csv(filepath, index=False)
```

7.2 Data preprocessing with features extraction code

```
import pandas as pd
import mne
import numpy as np
import matplotlib.pyplot as plt
from biosppy.signals.eeg import eeg
from biosppy.signals.tools import windower
import collections
import os

def _calc_vals(vals):
    c = collections.Counter(vals)
    if c[0.0] >= len(vals)/2:
        return 0
    return 1

for oo in range(13, 14):
    n = str(oo)
    n = n.zfill(2)
```

```
print("Working With chb", n)
if not os.path.exists("results/chb%s" % n):
    os.mkdir("results/chb%s" % n)
d = pd.read_csv("results/chb%s_res.csv" % n, index_col=0)
for col in d.columns:
    print("Working With chb", n, col)

    fileName = "path_to_data_file\chb%s\%s" % (n, col.strip())
    data2 = mne.io.read_raw_edf(fileName)
    data1 = data2.get_data()
    data1 = np.array(data1).T
    data1_res = np.array(d[col])
    sampling_rate = 256
    data1_res = [x for x in data1_res if str(x) != "nan"]
    length = len(data1_res)
    overlap = 0.5
    wind_size = 0.25
    wind_size = int(wind_size * sampling_rate)
    step = wind_size - int(overlap * wind_size)
    nb = 1 + (length - wind_size) // step
    r = []
    index, values = windower(signal=data1_res,
                             size=wind_size,
                             step=step,
                             kernel='hann',
                             fcn=_calc_vals,)
    data1 = np.array(data1)[:, 0:23]
    res = eeg(signal=data1, sampling_rate=sampling_rate, show=False)

    theta = np.array(res["theta"])
    alpha_low = np.array(res["alpha_low"])
    alpha_high = np.array(res["alpha_high"])
    beta = np.array(res["beta"])
    gamma = np.array(res["gamma"])

    in_vals = {'gamma': gamma, 'beta': beta,
               'alpha_high': alpha_high, 'alpha_low': alpha_low, 'theta': theta}
    final_vals = {'gamma': [], 'beta': [],
                  'alpha_high': [], 'alpha_low': [], 'theta': []}
    for k in final_vals.keys():
        for j in range(0, len(gamma)):
            tmp = 0
            for i in range(0, len(gamma[0])):
                tmp = (tmp + in_vals[k][j, i])/2.0
            final_vals[k].append(tmp)

    final_vals['result'] = values
    df = pd.DataFrame(final_vals)
    filepath = 'results/chb%s/%s_bans_excel_file.csv' % (
        n, col.strip().split('.')[0])
    df.to_csv(filepath, index=False)
```

7.3 CNN code for raw data

```
import os
import pandas as pd
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from tensorflow.keras import layers
import tensorflow as tf
import numpy as np
import keras
import json

from keras.layers.advanced_activations import LeakyReLU
from tensorflow.python.keras.utils.tf_utils import to_numpy_or_python_type
f_res = {}
for fi in range(1, 11):
    ff = str(fi)
    ff = ff.zfill(2)

    ff_name = "chb%s" % ff
    print("start with", ff_name)
    if not os.path.isfile("results_no_band/%s/resf/%s.csv" % (ff_name, ff_name)):
        continue
    df = pd.read_csv("results_no_band/%s/resf/%s.csv" %
                (ff_name, ff_name))
    f_res[ff_name] = {}
    dataset = df.values
    X = dataset[:, 0:23]
    Y = dataset[:, 23]

    batch_size = 20
    epochs = 30
    num_classes = 2

    min_max_scaler = preprocessing.MinMaxScaler()
    X_scale = min_max_scaler.fit_transform(X)
    # print(X_scale)
    max1 = 0
    avg = 0
    tp = 0
    tn = 0
    fp = 0
    fn = 0
    f = 0
    print("Working with %s" % ff_name)
    for kka in range(1, 11):
        print("try number: ", kka)
        fashion_model = Sequential()
```

```
fashion_model.add(layers.Conv1D(filters=64, kernel_size=3,
                        activation='relu', input_shape=(23, 1), padding='same'))
fashion_model.add(LeakyReLU(alpha=0.02))
fashion_model.add(layers.AveragePooling1D(pool_size=3, padding='same'))
fashion_model.add(LeakyReLU(alpha=0.02))
fashion_model.add(layers.LSTM(120))
fashion_model.add(LeakyReLU(alpha=0.02))
fashion_model.add(layers.Dense(200, activation='relu'))
fashion_model.add(layers.Flatten())
fashion_model.add(LeakyReLU(alpha=0.02))
fashion_model.add(layers.Dense(1, activation='sigmoid'))
m = [tf.keras.metrics.AUC(), tf.keras.metrics.TruePositives(), tf.keras.metrics.TrueNegatives(
), tf.keras.metrics.FalsePositives(), tf.keras.metrics.FalseNegatives()]
fashion_model.compile(loss=keras.losses.binary_crossentropy,
                optimizer='adam', metrics=[m])

X_train, X_val_and_test, Y_train, Y_val_and_test = train_test_split(
    X_scale, Y, test_size=0.3)
X_val, X_test, Y_val, Y_test = train_test_split(
    X_val_and_test, Y_val_and_test, test_size=0.5)
newX_train = tf.reshape(X_train, (len(X_train), 23, 1))
newY_train = tf.reshape(Y_train, (len(Y_train), 1))
newX_test = tf.reshape(X_test, (len(X_test), 23, 1))
newY_test = tf.reshape(Y_test, (len(Y_test), 1))
newX_val = tf.reshape(X_val, (len(X_val), 23, 1))
newY_val = tf.reshape(Y_val, (len(Y_val), 1))

train_data = tf.data.Dataset.from_tensor_slices(
    (newX_train, newY_train))
valid_data = tf.data.Dataset.from_tensor_slices((newX_val, newY_val))
fashion_train = fashion_model.fit(newX_train, newY_train,
                        batch_size=batch_size, epochs=epochs, verbose=0,
                        validation_data=(newX_val, newY_val))

test_eval = fashion_model.evaluate(newX_test, newY_test, verbose=0)
if not f:
    _, avg, tp, tn, fp, fn = test_eval
    f = 1
    max1 = test_eval
else:
    _, avgt, tpt, tnt, fpt, fnt = test_eval
    avg = (avg + avgt)/2.0
    tp = (tp + tpt) / 2.0
    tn = (tn + tnt) / 2.0
    fp = (fp+fpt) / 2.0
    fn = (fn+fnt)/2.0
met = [avg, tp, tn, fp, fn]
if test_eval[1] > max1[1]:
    max1 = test_eval
print('Test loss:', test_eval[0])
print('Test accuracy:', test_eval[1])
```

```
    f_res[ff_name]["Max_acc"] = max1
    f_res[ff_name]["met"] = met
    print("Max acc is: ", max1)
    print("Avarage: ", met[0])
with open("results_no_band/fineRes.txt", "w+") as file_j:
    file_j.write(json.dumps(f_res))
df = pd.DataFrame(f_res)
filepath = 'results_no_band/finalRes.csv'
df.to_csv(filepath, index=False)
```

7.4 Local Data Collection

Arab American University
Deanship of Scientific Research
Tel: 04-241-8888, ext 1196
mail: src@aaup.edu

الجامعة العربية الأمريكية
عمادة البحث العلمي
هاتف 04-241-8888, ext 1196
البريد الالكتروني src@aaup.edu

SRD:19/20-O-063

16/7/2021

حضرة د. عدنان فرهود المحترم
مدير مستشفى المقاصد/ القدس

تحية طيبة وبعد،

الموضوع: تسهيل مهمة بحثية

تهديكم الجامعة العربية الامريكية أطيب التحيات، وبالإشارة الى الموضوع أعلاه يرجى من حضرتكم التعاون بتسهيل المهمة البحثية لطالب الماجستير في علم الحاسوب " كريم خليل" ليتمكن من الحصول على بيانات تخطيط الدماغ EEG لـ 5-10 مرضى، من خلال قسم العلوم العصبية وبالتعاون مع د. مطيع الأشهب/رئيس القسم؛ حيث سيقوم الطالب باستخدام البيانات لأغراض البحث العلمي فقط. وذلك باستخدام خوارزميات الذكاء الاصطناعي لتصنيف إشارات تخطيط الـ EEG .

المعلومات المطلوبة: - تخطيط EEGالـ 5-10 مرضى ممن تظهر عليهم حالات تشنج أثناء اجراء التخطيط وبدون أي معلومات شخصية لأي مريض فقط الجنس والعمر.

شاكرين لكم حسن تعاونكم،،،

عميد البحث العلمي

د. محمد عوض
عمادة البحث العلمية
DEANSHIP OF SCIENTIFIC RESEARCH

الملخص

يمكن اعتبار جسم الإنسان كمجموعة من الأنظمة الفرعية التي تتواصل مع بعضها البعض، والدماغ هو وحدة التحكم في هذه الأنظمة. يعمل الدماغ من خلال مجموعة من الأجزاء الصغيرة تسمى الخلايا العصبية ، حيث ترتبط مجموعة من الخلايا العصبية لإنشاء شبكة عصبية لأداء بعض المهام. عندما تواجه هذه الخلايا العصبية مشاكل أو تعطي إشارات خاطئة ، سيبدأ الدماغ في إرباك أجهزة الجسم ويمكن أن يؤدي ذلك إلى مشاكل عصبية مثل الصرع. لفهم وتشخيص مثل هذه المشكلة يتم إجراء اختبار يسمى مخطط كهربية الدماغ (EEG)، حيث يسجل الأنشطة الكهربائية للدماغ بحيث يمكن أن يشير كل جزء من هذا السجل إلى نشاط أو مجموعة من الأنشطة المتعلقة بالدماغ. تكمن مشكلة مخطط كهربية الدماغ في أنه ينتج قدرًا هائلاً من البيانات في فترة تسجيل قصيرة ، وبسبب تعقيد إشارات الدماغ والأخطاء البشرية ، يستغرق الأطباء الكثير من الوقت لإجراء التشخيصات ، ويتم تشخيص العديد من المرضى بشكل خاطئ. هذا يؤدي إلى الحاجة إلى وجود نظام محوسب يمكنه تقليل هذه المشاكل.

تم اقتراح العديد من الأنظمة خلال السنوات السابقة ، ومع مزايا الذكاء الاصطناعي والتعلم الآلي ، تم إجراء الكثير من الأبحاث في هذا المجال. تم إنشاء بعض التطبيقات باستخدام أنظمة قائمة على القواعد ، والبعض الآخر باستخدام .Multilayer Perceptron (MLPs) عندما بدأت شبكات التعلم العميق مثل الشبكات العصبية التلافيفية (CNN) في الانتشار ، تم إنشاء العديد من التطبيقات أيضًا بناءً عليها. المشكلة في فحص اشارات الدماغ(EEG) هي أن الأنماط ليست فريدة بالضرورة ؛ حيث يمكن أن تعني الإشارة نفسها لمرضى مختلفين أشياء مختلفة ، مما يجعل من الصعب جدًا إنشاء نموذج عام لمعالجة إشارات الدماغ.

في هذا البحث ، تم اقتراح نهج جديد حيث نستفيد من قدرات CNN لاستخراج الاشارات المميزة والتعامل مع اشارت الدماغ المعقدة، ودمجها مع طرق تحليل الإشارات الموجة ، وخطوات معالجة أخرى لإنشاء نموذج قوي لتحليل بيانات الدماغ و الكشف عن نوبات الصرع. تكمن القوة الرئيسية لعملنا في أنه مبني على مستوى المريض ؛ نظرًا لأن اختبار EEGيوفر كمية هائلة من البيانات ، فمن الممكن ضبط النموذج لكل مريض. في عملنا ، أنشأنا نموذج (1D CNN) مع طبقات Conv1D وذاكرة طويلة المدى (LSTM) وطبقات .MLPS استخدمنا مجموعة قراءات عالمية للعديد من المرضى الذين يعانون من الصرع. في هذا البحث ، قمنا بمقارنة عملنا مع الخوارزميات العادية الأخرى مثل (Regression Tree)و K-Nearest Neighbours (KNN) و (SVM) و .(Ensemble)تعمل خوارزمية المعالجة الاولية للبيانات لدينا على تحسين البيانات وجعلها أفضل ، حيث قدم النموذج المقترح دقة كشف أفضل بكثير عند استخدامه مع البيانات التي تمت معالجتها ، وكانت الخوارزميات العادية تفعل ذلك بناءً على الدراسات السابقة التي راجعناها. تمكنا من الحصول على متوسط دقة كشف إشارات الصرع تبلغ 97.14٪ ، بأعلى دقة 99.2٪. في الأصل كان هذا البحث يستهدف البيانات المحلية الفلسطينية ، وللأسف لم يكن هذا النوع من البيانات متاحًا ، ونتيجة لذلك تم استخدام مجموعة قراءات عالمية