PERFORMANCE COMPARISON OF SCHIZOPHRENIA DETECTION USING EEG SIGNALS

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE

OF

MASTER OF TECHNOLOGY

IN

SIGNAL PROCESSING AND DIGITAL DESIGN

Submitted by:

HARIT KANT SHIVA

2K20/SPD/06

Under the Supervision Of

Dr. N. Jayanthi

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

MAY, 2022

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, HARIT KANT SHIVA, Roll No. 2K20/SPD/06 student of M.Tech (SIGNAL PROCESSING AND DIGITAL DESIGN), hereby declare that the project Dissertation titled "**PERFORMANCE COMPARISON OF SCHIZOPHRENIA DETECTION USING EEG SIGNALS**" which is submitted by me to the Department of ELECTRONICS & COMMUNICATION ENGINEERING, Delhi Technological University, Delhi in partial fulfillment ofthe requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associate ship, Fellowship or other similar title or recognition.

Place: Delhi **HARIT KANT SHIVA**

Date: 26-05-2022

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the major project titled "**PERFORMANCE COMPARISON OF SCHIZOPHRENIA DETECTION USING EEG SIGNALS**" which is submitted by HARIT KANT SHIVA, Roll No-2K20/SPD/06 ELECTRONICS & COMMUNICATION ENGINEERING Department, Delhi Technological University, Delhi, in partial fulfilment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or fullfor any Degree to this University or elsewhere.

Place: Delhi

Date: 26-05-2022

Dr. N. Jayanthi

(SUPERVISOR)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I am highly grateful to the Department of Electronic & Communication Engineering, Delhi Technological University (DTU) for providing this opportunity to carry out this project work. I owe special debt of thankfulness and gratitude towards my supervisor **Dr. N. Jayanthi** of Department of Electronics & Communication Engineering, Delhi Technological University, Delhi for guiding me, helping and inspiring me with suggestions in my present work and is acknowledged with reverential thanks.

Finally, I would like to expresses gratitude to all faculty members of Electronic & Communication Engineering Department, DTU for their intellectual as well as moral support during my M.Tech study at DTU.

Place: Delhi **HARIT KANT SHIVA**

Date: 26-05-2022

ABSTRACT

Schizophrenia is a kind of mental illness that affects about 1% of the world's population. It may be difficult for schizophrenic patients to distinguish between internally generated and externally generated stimuli due to one or more problems with the nervous system's corollary discharge process. This is one explanation for some symptoms of schizophrenia. As a result, looking into this process and how it relates to the disease's symptoms could help us learn more about the abnormal brain processes seen in patients with this disease. New avenues in the study of electrophysiological brain activity can be explored, and ambulatory neuronal disease diagnosis can be performed, thanks to improved access to EEG data. The aim is to use a variety of diagnostic techniques to identify neuronal pathologies. The dataset used to test the methods was a larger sample replication of EEG data from previous studies published on July 10, 2013 in Schizophrenia Bulletin Advance Access. The electroencephalogram (EEG) data of 22 healthy people and 36 people with schizophrenia were combined with the EEG data of 10 healthy people and 13 schizophrenia patients from a previous study. Two different methods for diagnosing SZ using EEG signals were considered during the classification phase. To classify EEG signals, traditional machine learning techniques have been used. Among the methods used were xgboost, decision tree, naïve bayes, random forest, Long short-term memories (LSTMs), support vector machines, two-dimensional convolutional networks (2D-CNNs), and two-dimensional convolutional networks-LSTMs. The Deep Learning models were implemented at this point, and a variety of activation functions were compared. Among all proposed models, the SVM architecture has demonstrated the highest level of performance. The RBF Kernel with Cross Validation(CV) = 3, 6, 8 are used in this architecture. A precision of 100 percent is achieved using the SVM model.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

EEG data is becoming more accessible and accurate, providing unprecedented opportunities to better comprehend brain activity, contribute to the development of individualised prognoses, as well as promotes health using the use of various type of biofeedback systems [1]. Electroencephalography, unlike other techniques based on brain imaging like MRI, has a high temporal content in addition to being non-invasive, safe, and inexpensive. Imaging by magnetic resonance, for instance, incurs increased costs and risks, in addition to recording frequency restrictions [2]. Electroencephalograms (EEGs) have been demonstrated to be an effective research tool for schizophrenia, which affects approximately one percent of the global population and is frequently misdiagnosed [3]. Due to the small size of populations [4] as well as the inherent difficulty of extracting brain signals, monitoring electrophysiological brain activity has a limited application in the diagnosis of neurodegenerative diseases. Case-control populations are composed of patients and healthy individuals serving as controls. Obtaining reliable results using case-control populations is also difficult. This remains true despite the fact that monitoring this activity has numerous benefits apart from those derived from monitoring the activity itself. [5] Brain data is multivariate, high-dimensional, temporally-spatially-spectrally dense, and susceptible to artefacts and noise. Furthermore, the data varies greatly between individuals. Researchers examined the corollary discharge in individuals with schizophrenia and healthy controls by having them perform a simple button-pressing task. Participants either actively generated a tone by pressing a button that generated it instantly, sat back and listened to the same tone, or actively generated no tone by pressing a button that did not generate a tone. The N100, a negative deflection of the EEG brain wave that occurs 100 milliseconds after a sound's onset, was not suppressed when schizophrenia patients generated a tone as opposed to passively listening to it. This discovery was made possible by the active participation of schizophrenia patients in the experiment. This occurred as a direct result of the level of patient participation in the study.

In conventional machine learning, selection for the SZ appropriate algorithm for feature extraction of the diagnosis is a challenging job demanding extensive signal processing and expertise in the field of AI. To successfully complete the task, conventional machine learning employs supervised learning. Both will be necessary for the successful completion of this task. To achieve the desired results, the successful completion of this task requires both of these

elements. In recent years, EEG-based DL methods for diagnosing schizotypal personality disorder have been developed. Using these methods, the extraction of features is performed in a manner distinct from the construction of the deep layer. [6]. Shalbaf et al. [7] propose an electroencephalogram (EEG)-based model on transfer learning for the schizophrenia diagnosis. In this study, EEG signals were processed using the ResNet18 model so that features could be extracted from the underlying data. In addition, SVM was utilised from the beginning to the end of the classification phase of the project. Diverse CNN models have been investigated to determine the extent to which they might be useful for SZ diagnosis based on EEG. Applying CNN models to the SZ diagnostic procedure produced positive outcomes in [8] and [9]. Convolutional recurrent neural network (CNNRNN) models are used to diagnose a variety of brain diseases [8, 9]. Convolutional neural network with recurrent connections is denoted by [CNNRNN]. CNNRNN models are frequently referred to by the abbreviation "CNNRNN." In [10], [11], [12], and [13], CNN-LSTM models were integrated into the SZ diagnostic procedure, with encouraging results.

Using the dataset published by Brian Roach (https://www.kaggle.com/datasets/broach/button-tone-sz) , the SVM, CNN, LSTM as well as the CNN+LSTM models are being trained. Before concluding that one IMF's findings are more accurate than those of other IMFs, we will examine the correlation between the two different data sets.

The thesis is supported by the following outline. In the first section of Chapter 2, the current contributions to individual diagnosis based on brain signal data are discussed in greater detail in the form of Literature Review. In the third chapter of this thesis, the problem is presented in a more formal manner, followed by a summary of the key concepts that must be understood in order to proceed. In the fourth chapter, we provide a comprehensive review of the suggested procedures. In the fifth chapter, you will find a thorough domain analysis of the EEG signal dataset as well as a demonstration of the machine learning and DL methods that can be used to diagnose schizophrenia. In Chapter 6, we discuss the Results of our proposed methods and which method actually worked for us followed by the chapter 7 which contains the conclusion of the closing remarks and future scope.

CHAPTER 2

LITERATURE REVIEW

The authors of "The effect of individual alpha neurofeedback training on short-term memory" are Wenya Nan, Joo Pedro Rodrigues, Jiali Ma, Xiaoting Qu, Feng Wan, Pui-In Mak, Peng Un Mak, Mang I Vai, and Agostinho Rosa. Each of these authors contributed in some way to the book. Higher alpha activity levels on an electroencephalogram (EEG), which researchers used to measure brain activity, were associated with the ability to recall information. By training with individual alpha neurofeedback, individuals could enhance their neuronal short-term memory (NFT) (NFT). During their time with NFT, participants learned how to increase the relative amplitude of their own alpha band by employing a technique developed specifically for NFT. After participating in 20 NFT sessions, they were significantly better able to recall recent memories. In addition, additional research has demonstrated that training increases the relative amplitude of a person's upper alpha band, which is associated with enhanced short-term memory. [Requires citation] [Requires citation] In order to demonstrate this, the participants' brain activity was monitored. Researchers also discovered a link between positive thinking and the most effective mental strategies for individual alpha training.

Adrian J. Fowle and Colin D. Binnie published in their paper findings, "EEG Uses and Misuses in Epilepsy," that there is little evidence-based medical literature on the electroencephalogram (EEG) is both unexpected and disheartening (EBM). Given the recent emphasis on evidence-based medicine, this finding is unexpected (EBM). Even though there is a Cochrane review group dedicated to the study of epilepsy, it has not yet conducted research on EEG as a subject in and of itself. It appears that Medline, which many people use as their primary source of scientific information, categorises EEG-related papers incorrectly. Even though "EEG" appears in the title, this remains true. Both Cochrane and Medline contain tens of thousands of citations to articles discussing how drugs affect the EEG and how various disorders manifest in the EEG. According to the principles of evidence-based medicine, these publications must support the use of EEG with evidence. In many parts of the world, there is insufficient evidence to support the use of the EEG, so it is possible that some individuals will incorrectly believe that the EEG is ineffective. In addition, medical care is extremely costly in these regions. The electroencephalogram, or EEG, is an effective method for diagnosing epilepsy. However, improper usage is possible. In some instances, the electroencephalogram (EEG) is an effective method for diagnosing epilepsy. Because of this, obtaining an EEG in such a setting is

frequently viewed as abusive. Ictal recording is the only way to distinguish between an epileptic seizure and a non-epileptic seizure in a patient with unexplainable recurrent symptoms. This can be accomplished through EEG telemetry or recording while the subject is in motion. However, once epilepsy has been identified, the electroencephalogram (EEG) is likely to be the most useful diagnostic tool for determining the type of epilepsy, the prognosis, and the most effective treatment. Electroencephalograms, or EEGs, are the most effective method for determining the location of partial seizures. They are also an integral part of the evaluation process for people who are considering epilepsy surgery. EEG monitoring of how epilepsy worsens is ineffective unless the doctor can observe changes. Also, the EEG is not used to determine the efficacy of antiepileptic drugs (AEDs), despite the fact that it could be useful for determining whether they are dangerous. The EEG can help determine if it is safe to discontinue the use of AEDs on children. Regarding adults, the electroencephalogram, or EEG, is significantly less important in making this determination. Evaluating how well an EEG would function in a given environment is a difficult task that should only be undertaken by someone with prior experience with the technology. For any EEG interpretation, the clinical picture (EEG) must be considered (EEG). There is a possibility that the difficult decisions discussed in this article will make it more difficult to comprehend the EEG exam report. It is crucial that the EEG department and the physicians who send patients there can communicate for a number of reasons. This holds true for both individual testing and staff training.

A. Shalbaf, S. Bagherzadeh, and A. Maghsoudi presented a paper titled "Transfer learning using deep convolutional neural network for automated identification of schizophrenia from eeg signals" at the Physical and Engineering Sciences in Medicine conference in 2020. According to the findings of this study, schizophrenia is a severe brain disorder that can impede a person's ability to think, remember, comprehend, communicate, and perform a variety of other daily tasks. If a person with schizophrenia is not promptly diagnosed and treated, their odd behaviour can worsen over time. So, detecting SZ early may aid in its treatment or prevention. Electroencephalography, also known as EEG, is frequently used to study brain disorders such as schizophrenia due to its low cost and high temporal resolution. In this study, an automatic method for distinguishing between people with schizophrenia and healthy controls is proposed. Transfer learning is used in addition to deep convolutional neural networks (CNNs) to achieve this aim (CNNs). The time-frequency algorithm, also known as the continuous wavelet transform (CWT), is employed to generate images from EEG signals. Four prominent CNNs have been trained to recognise images of EEG signals. These CNNs are

named AlexNet, ResNet-18, VGG-19, and Inception-v3, in that order. The SVM classifier receives the results of these models' convolutional and pooling layers. These characteristics are also known as deep qualities. We altered the classification parameters of the SVM so that we could distinguish between healthy individuals and SZ patients. The EEG readings of 28 people, including 14 healthy volunteers and 14 people with SZ, are analysed to determine whether or not the proposed approach is effective. Using the frontal, central, parietal, and occipital regions with the ResNet-18-SVM yielded the highest levels of accuracy, sensitivity, and specificity, with values of 98.60 percent 2.29, 99.65 percent 2.35, and 96.92 percent 2.25, respectively, for accuracy, sensitivity, and specificity. As a result, the methodology could be utilized as a diagnostic means to help physicians quickly identify and treat SZ patients.

In 2019, Application Sciences published the paper which 2,870 readers accessed. This article discusses a computerised method for diagnosing schizophrenia. Utilizing SZ convolutional neural networks (SZ). Schizophrenia is a brain disorder that, among other symptoms, causes disorganised speech and auditory hallucinations. Schizophrenia can also cause individuals to believe falsehoods. Electroencephalograms, also known as EEGs, are frequently employed in research and the diagnosis of brain-related diseases. The EEG signals were analysed using a CNN with eleven layers. This model examined EEG signals from 14 healthy volunteers and 14 individuals with SZ. Traditional machine learning algorithms are susceptible to differences between observers and require extensive training time to function effectively. In this area of study, deep learning strategies are employed to facilitate the automatic extraction and categorization of relevant data. The convolution stage is responsible for automatically extracting features. In contrast, the max-pooling stage is responsible for extracting the most important features. The fully linked layer is the most effective method for sorting signals into distinct groups. The classification accuracy of the proposed model was 98.07 percent when it was used for non-subject-based testing, but it dropped to 81.26 percent when it was used for subject-based testing. Clinicians can utilise the proposed model to diagnose SZ in its earliest stages.

Carlos Alberto Torres Nairo and Cristian Jos'e L'opez Del Alamo co-authored an article that was published in the Volume 10 Issue 10 of the International Journal of Advanced Computer Science and Applications (IJACSA) (IJACSA). According to the findings of this study, schizophrenia affects more than 21 million people worldwide. People with severe mental illnesses are frequently mistreated, stigmatised, and denied their human rights. Signals from an electroencephalogram (EEG) reveal how the brain functions and how diseases alter this

activity. Consequently, they are used to categorise and diagnose mental disorders. This allows for more precise results. This study presents a model for grouping schizophrenia patients and healthy individuals. The model was built using EEG signals and Deep Learning techniques. The model is based on both the data provided by EEG signals and the efficiency demonstrated by Deep Learning algorithms. Due to the EEG's high dimensionality and the number of channels, the Pearson Correlation Coefficient (PCC) was chosen to demonstrate the relationships between the channels (PCC). We therefore did not use an EEG to feed CNN with a massive amount of data; rather, we fed it a smaller matrix (CNN). The research demonstrated that the classification model derived from EEG data was 90% accurate, 90% specific, and 90% sensitive.

In the volume11, number1 issue of Scientific Reports in 2021, J. Sun, R. Cao, M. Zhou, W. Hussain, B. Wang, J. Xue, and J. Xiang published "A hybrid deep neural network for categorising schizophrenia with the help of EEG data". According to the research findings discussed in this article, it is a very severe mental illness that causes its sufferers great suffering. Due to this, it is crucial to obtain an accurate diagnosis as soon as possible. This research has focused to improve the classification accuracy of electroencephalography (EEG) signals in both individuals with schizophrenia and healthy controls. This was accomplished by locating and identifying an improved method of describing electroencephalography (EEG) signals. Our method of instruction consists of two distinct components. A series of spatially informative red, green, and blue (RGB) images are generated from these characteristics. Fuzzy entropy (also known as FuzzyEn) appears to be more significant in brain topography than fast Fourier transform (also known as FFT) (also known as FFT). The proposed deep learning (DL) method has an accuracy of 99%, while FFT as well as the FuzzyEn have an accuracy of 96% and 99.22 percent, respectively. Using a hybrid DNN after obtaining fuzzy features from an EEG time series as input features is the optimal method for achieving accurate classification, based on these findings. It was determined that this was the optimal approach. In this field, significant progress has been made in comparison to the most cutting-edge techniques currently employed.

In the 2021 edition of Biomedical Signal Processing and Control, volume 66, the authors of this study A. M. Joshi, A. Sharma, and A. Parashar published a paper based on electroencephalograms, "Dephnn: A Novel Hybrid Neural Network for Depression Screening" also known as "A Novel Hybrid Neural Network for Depression Screening" (A Novel Hybrid Neural Network for Depression Screening). Depression is distinct from other mental disorders because it causes persistent sadness. It is essential to understand that this disease can affect

people of all ages and from all over the world. People anticipate that early detection of this disease, which is currently considered a global threat, will save many lives. Electroencephalogram (EEG) signals can reveal how a person is currently thinking, which can assist in diagnosing a mental disorder. This article examines and discusses the advantages of a fully automated Depression Detection System. This is because manually analysing EEG data is extremely difficult, time-consuming, and requires a great deal of prior knowledge. DepHNN (Depression Hybrid Neural Network) is a new EEG-based computer-aided (CAD) Hybrid Neural Network that can be used to screen patients for depression, according to the findings of this study. Convolutional neural network (CNN) architectures are utilised for temporal learning, windowing, sequence learning, and long-short term memory (LSTM) in the proposed method (LSTM). Neuroscan was used to collect EEG (electroencephalogram) data from 24 healthy people and 21 depressed people who did not take drugs for this model. Compared to other methods, the windowing method saves the model a great deal of time and effort. The precision is 99.10%, while the mean absolute error (MAE) is 0.204%. The CNN-LSTM hybrid model developed to detect depression from EEG signals was found to be accurate, user-friendly, and effective.

Kuldeep Singh, Jytirmesh Malhotra and also Sukhjeet Singh published a paper entitled "Spectral characteristics-based CNN for accurate as well as the rapid detection of schizophrenia patients" in the year 2020. Schizophrenia is an incurable mental disorder, according to the information in this article. This condition affects millions of people worldwide and causes them to think, feel, and act in a peculiar manner. In this era of the internet of things, cloud computing, and machine learning, it is difficult to overstate the importance of being able to diagnose schizophrenia using a computer. The objective is to assist individuals with schizophrenia in leading happier, more fulfilling lives. Utilizing spectral features and a model known as a convolutional neural network, or CNN, this study demonstrates how to accurately identify patients suffering from schizophrenia. Throughout the entirety of the model's development, multichannel EEG recordings were analysed in real-time. In this model, EEG signals are processed using techniques such as filtering, segmentation, and frequency domain conversion. During this process, the frequency domain segments are transformed into the spectral bands of the frequencies delta, theta-1, and theta-2, as well as alpha, beta, and gamma. Formulas for calculating the mean spectral amplitude, spectral power, and Hjorth descriptors are derived using the spectral characteristics of each band (Activity, Mobility, and Complexity). The convolutional neural network (CNN) and the long short-term memory network (LSTM)

classification models acquire these spectral characteristics in distinct ways. Raw time-domain and frequency-domain EEG segments can be classified using CNN models with comparable architectures to those described in this paper. The proposed spectral features-based convolutional neural network (CNN) model was found to be an efficient and accurate method for identifying schizophrenic patients among healthy individuals. This was determined by examining the simulation results for each model. Using the appropriate classification times, this method achieves an average classification accuracy of 94.08 and 98.56 percent for two distinct datasets, respectively. Moreover, this method produces the quickest classification times.

Siuly Siuly, Smith K. Khare, Varun Bajaj, Hua Wang, and Yanchun Zhang presented in their article "A Computerized Method for Automatic Detection of Schizophrenia Using EEG Signals" an EMD-based method for identifying SZ patients based on EEG signals. This method was used to identify patients with schizophrenia. EMD can convert a fixed number of nonlinear and nonstationary EEG signals into a fixed number of IMFs. In order to discover crucial information in the time domain, a signal processing system requires constant, unchanging signals. After collecting these five statistical characteristics from each IMF, a KW test was used to determine how well each was able to distinguish itself from the others. According to the researchers, IMF 2 outperformed the other IMFs when classifying SZ and HC EEG signals. Their method paved the way for the development of new techniques for analysing and detecting SZ using EEG signals, which are currently difficult to classify due to their nonlinear and dynamic characteristics.

This paper's authors are also the authors of a 2017 paper titled "Automatic Schizophrenia Diagnosis Using EEG Signals Modelling with CNN-LSTM". Researchers utilised a variety of traditional machine learning-based classification algorithms when attempting to diagnose SZ by analysing EEG signals. The algorithms used the normalised versions of the EEG signals as classification features. This research's proposed model is intended to be more precise than models proposed for the vast majority of other studies. The proposed model, which uses EEG signals as a diagnostic aid, can be implemented on specialised software and hardware platforms to aid hospitals in the rapid diagnosis of SZ 10 cases.

CHAPTER 3

BACKGROUND OF EEG

In the section, we begin with explanation of the dataset that was used for analysis of our study, as well as the pre-processing procedure used. Following that, we discuss the methodology plan that has been proposed. The framework of the projected methodologies for classifying SZ as well as the HC based on signals of the EEG is depicted in Figure 1. [9]. The method that has been suggested includes the following procedures:

i. Calculating statistical characteristics and shrinking the dimensions of the IMFs are two of the steps involved.
ii. Generation of a feature matrix that combines significant features using the EMD algorithm.
iii. EEG signal decomposition into AM-FM components (referred to as IMFs).
iv. categorising the compiled feature matrix by employing multiple classifiers.

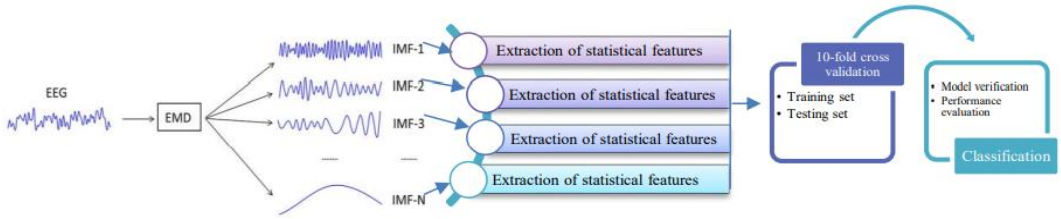

Fig.1. Identifying SZ patients using EEG signals using the proposed framework.

3.1 Electroencephalography

Electroencephalography, or EEG, measures the sum of inhibitory as well as the excitatory postsynaptic potentials [5]. This information reflects the electrophysiology of the brain. Individual neurons generate electrical fields that, when combined, produce externally observable effects. EEG signals are less expensive and easier to acquire than other brain imaging techniques, making them more desirable for use. This increases the attractiveness of their application. Even though there is a great deal of noise and the spatial resolution is poor, the temporal resolution is excellent. $\vec{x}_{i\in 0..m}$ is a type of time series, which is a collection of measured observations overtime, with every observation possessing its own set of characteristics $\vec{x}_i = x_i^0, \dots, x_i^p$. The definition of a time series is a collection of observations. If $p = 1$, it is known as the univariate and it is known as the multivariate if $p > 1$. If the characteristics are numeric, their values are real; if they are nominal, their values are symbolic.

Each observation in a tabular dataset consists of a collection of characteristics. In contrast, each observation in a time series-based dataset consists of a time series.

3.1.1 Frequency bands

The frequency domains of EEG signals are extremely important due to the fact that they can be interpreted as a collection of waves produced by a variety of neuronal ensembles. Studies of EEG spectral values using statistical factor analysis produce clusters of frequency components that display significant overlap with frequency bands that are already well known. Infralow (less than 0.2 Hz), δ (ranging between 0.2 Hz-3.5 Hz), θ (ranging between 4 Hz to 7.5 Hz), α as well as μ (ranging between 8 Hz -13 Hz), β (ranging b/w 14 Hz to 30 Hz), γ (ranging between 30 Hz to 90 Hz) [5].

3.1.2 EEG Properties

The authors in the article [5] describe in detail how the impact of this signal has been observed in the many different subfields of neuroscience research. Within the frequency band that is being investigated, there needs to be a spectral peak for it to be possible to identify an EEG oscillation. One of its characteristics is the maximum frequency that the oscillation can reach, in addition to its power as well as the bandwidth. The authors contend that a collection of neurons would be able to function as a single entity, which necessitates a population of neurons that is sufficiently large, temporally and spatially organised, and whose electrical fields can be recorded at a distance. The evidence suggests that the brain is made up of distinct communities that are responsible for information processing and communication. A complete understanding of cognitive processes requires familiarity with the data and communication processing that occurs within the brain [5].

3.1.3 EEG channels

The EEG signal is typically measured using electrodes placed in a variety of positions, it can be thought of as a multivariate time series. The collection of electroencephalographic data that comes from a single location is referred to as a channel. The identifier of a channel is the geographical region in which the channel can be found. The nomenclature of each channel as well as its location is shown, for instance, in Figure 2.1. The first few letters of the EEG identifier provide information about the general location of the lobe, while the remaining characters provide information about a specific location within the lobe. The frontal lobe channels are represented by the letters $F3, F4, F7, F8, Fp1, Fp2, Fz$. The mid-scalp channels

are represented by the letters $C3, C4, and\ Cz$. The temporal lobe channels are represented by the letters $T3, T4, T5, and\ T6$. The parietal lobe channels are represented by the letters P3, P4, and Pz. The occipital lobe channels are represented by the letter O.

3.2 EEG Uses in Schizophrenia

There has been an extensive use of EEG imaging in the neural pathologies diagnosis, and it is a technique that is both appropriate and desirable for the schizophrenia pathology. This is because EEG imaging can detect subtle changes in brain activity that may not be detected by other diagnostic methods. According to the findings of [7], people who are diagnosed with schizophrenia have a response time that is noticeably more drawn out than healthy controls. The electroencephalogram (EEG) is a method that is frequently used for the diagnosis and investigation of schizophrenia, as indicated by a number of studies that are discussed in section 3. Both the ease with which EEG can be assessed and performed in virtually any psychiatric setting and the fact that it has been shown to be well tolerated by nearly all patients are important factors in its use in the diagnosis of schizophrenia. In addition, EEG has been shown to be accurate in predicting the prognosis of schizophrenia.

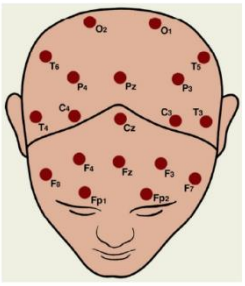

Figure 2.1: EEG channel mapping nomenclature (sourced rom [6]).

Fig 2. EEG Channel Mapping Nomenclature

[8] provides a concise summary of the EEG abnormalities that should be looked for in patients being evaluated for a diagnosis of schizophrenia. In order to accomplish this, they analysed the progression of spectral EEG deviations throughout the course of the study. Only research that compared the spectral power of schizophrenia patients to healthy controls as a control group was considered for inclusion in the meta-analysis. The existence of two groups (or populations), one with a healthy control group and the other with pathology, is required in order to classify distinguishing characteristics based on the collected signals. One of the groups must serve as the control group. It is necessary to choose one of these groups to serve as the control group. At one time, it was thought that schizophrenia was linked to increased levels of delta power, increased levels of theta power, decreased levels of alpha power, and increased levels

of beta power. In the wake of these preliminary discoveries, a number of follow-up investigations concluded that people with schizophrenia display noticeably higher levels of lower spectrum activity (slow waves). The most common type of slow wave abnormality, which manifests itself most frequently as an increase in delta frequency, is found in the frontal lobes. Electrode locations on an electroencephalogram (EEG) should ideally be reported when making a diagnosis of schizophrenia. These should include the frontal, temporal, central, parietal, and occipital electrodes.

Neural oscillations and the synchronisation of those oscillations are, according to event-related EEG traces, crucial devices for communication of intraneuronal processed by distributed brain regions [9]. The regulation of synaptic transmission between neurons is another process that is impacted by oscillations in the neural network. The use of EEG data analysis in schizophrenia research is supported by a number of previous EEG studies with the goal of gaining a more in-depth understanding of the pathophysiological processes that are responsible for the cognitive deficits that are characteristic of neuropsychiatric disorders.

3.3 Processing EEG signal

The representation of an EEG signal that is most commonly used is that of a multivariate asynchronous time series. If there is no discernible pattern throughout the entirety of a time series, then that time series may be considered to be asynchronous. During the course of an EEG recording session, multiple channels are utilised, which results in a signal that is composed of contributions from multiple sources. The signal will take on multivariate characteristics as a result of this. It has been suggested that the time domain of the EEG signal should be converted into the frequency domain so that spectral patterns can be analysed and then correlated with brain-evoked events. This would allow for greater accuracy in the analysis.

3.3.1 Fourier Transform

The Fourier transform is a crucial component of modern signal processing, and its namesake, Joseph Fourier, is credited with developing it. Because of this property, it is possible to segment a signal into waves, each of which possesses a frequency and intensity function of its own. The transformation known as the Fourier transform is used to represent a signal in the frequency domain. Because every signal in the real world is discrete, the Fourier Transform can only be applied to continuous signals (a collection of samples over time). Nevertheless, it is possible to convert continuous signals with its help.

3.3.2 Discrete Fourier Transform (DFT)

Utilization of the Fourier transform is commonplace in digital signal processing (discrete signals). The name of the Discrete Fourier Transform is represented by the symbol in question.

$$X_k = \sum_{i=0}^{N-1} x_i e^{-\frac{2\pi y_1}{N} ki}$$
$$k = 0, \ldots, N-1$$

X_k is the amplitude for the frequency k and there is a total of N frequencies.

3.3.3 Fast Fourier Transform

[10] suggested using the Fast Fourier Transform for a faster computation rather than the conventional Fourier Transform because the conventional Fourier Transform requires an excessive amount of computing power.

3.3.4 Short-time Fourier Transform (STFT)

The signal is first partitioned into windowed segments by the STFT, after which the Fourier transform is computed for each of these subsegments individually. Deciphering non-repetitive signals is a particularly strong suit of this method. This assumes that a periodic portion of a non-periodic signal can be identified by zooming in and examining it closely enough. This was derived from the fact that the above statement was true. This is the fundamental principle that underlies the STFT, and the primary reason why its application in EEG has become so widespread. The size of the window plays an essential role; the lower frequencies can be captured much more accurately by windows that are larger. Even when the frequency of the difference is low, it is still possible to find a difference between healthy controls and people with schizophrenia that is statistically significant.

CHAPTER 4

PROPOSED METHODOLOGIES

The primary purpose of this thesis is to categorise data from the brain in order to accurately diagnose the pathology associated with schizophrenia. The goal of the classification task is to correctly label new instances of unlabelled data by acquiring a mapping model between data and corresponding classes by making use of an annotated dataset as a source of information. The classification algorithms known as Decision Tree Classifier, Random Forest Classifier, Naïve Bayes, XGBoost, Support Vector Machines (SVM), Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and CNN-LSTM will be covered in this section.

This section provides insights about the methods used to classify our dataset namely DT, RF, NB, XGB, SVM, 2D-CNN, LSTM, and 2D-CNN-LSTM models for SZ diagnosis via EEG signals.

4.1 SVM (SVC and Linear SVM)

SVMs are a subgroup of SL techniques that can be used for both classification (SL or USL) and regression [2]. They are members of the family of generalised linear classification. Uniquely, the SVM algorithm reduces empirical classification error while simultaneously increasing geometric margin. As a direct result, Maximum Margin Classifiers were utilised by SVM. SVM employs the SRM algorithm, which is utilised to mitigate structural risk (SRM). In a higher-dimensional space, the support vector machine (SVM) generates a hyperplane that is maximally distinct from the input vector. The hyperplane sides on each end separating the data is composed of two parallel hyperplanes. "Separating hyperplane" refers to the hyperplane that creates the largest distance between two parallel hyperplanes. As the margin or distance between these parallel hyperplanes increases, it is assumed that the generalisation error of the classifier will decrease [2]. We consider data points of the form

$$\{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4) \dots\dots\dots, (x_n, y_n)\}.$$

Where y_n=1 / -1, a constant denoting the class to which that point xn belongs. n = number of sample. Each x n represents a p-dimensional real vector. Scaling is essential for protecting against variables (attributes) with greater variance. This Training data can be viewed by means of the dividing (or separating) hyperplane, which takes

$$w.x + b = 0$$

b is a scalar and w is a p-dimensional vector in this equation. The separation hyperplane is parallel to the vector, while the vector w is perpendicular to it. When the offset parameter b is utilised, the margin increases. If b is absent, the hyperplane must pass through the origin, which drastically reduces the number of possible solutions. We are particularly interested in SVM and parallel hyperplanes because we want to maximise our margin. An equation can be utilised to describe parallel hyperplanes.

$$w.x + b = 1$$

$$w.x + b = -1$$

If the training data can be separated linearly, we can select hyperplanes with no intermediate points and maximise the distance between them. According to geometry, the distance between the hyperplanes is equal to 2 w divided by the width. Consequently, we wish to keep w to a minimum.

$$w.\, x_i - b \geq 1 \quad \text{or} \quad w.\, x_i - b \leq -1$$

This is also possible as

$$yi\,(\,w.xi - b) \geq 1, \qquad 1 \leq i \leq n$$

Support vectors (SVs) are samples that run parallel to the hyperplanes (SVs). Support vectors are indicated by a separating hyperplane with the largest margin, as defined by M = 2 / w, that specifies support vectors. Which ones are adequate?

$$y_j\left[w^T \cdot x_j + b\right] = 1\,, i = 1$$

l represents the number of training data points. A machine learning algorithm should minimise $\|w\|^2$ while considering inequality constraints in order to find the optimal hyperplane with the largest margin.

$$y[w^T \cdot x_i + b] \geq 1; i = 1,2 \ldots 1$$

Lagrange's Function saddle points resolve this optimization issue.

$$L_p = L_{(w,b,c)} = 1/2 \parallel w \parallel 2 - \sum_{i=1} a_i\left(y_i\left(w^T x + b\right) - 1\right)$$

$$= 1/2\mathrm{w}^{\mathrm{T}}\mathrm{w} - \sum_{\mathrm{i}=1}^{1}\alpha_i\left(\mathrm{y}^{\mathrm{w}}\mathrm{w}^{\mathrm{T}}\mathrm{x}_{\mathrm{i}} + \mathrm{b}\right) - 1)$$

Lagranges must be minimised with respect to w and b and maximised with respect to nonnegative α_i ($\alpha_i \geq 0$), so the search for the optimal saddle point (w_0, b_i, α_0) is necessary. There are two ways to solve this problem: the primary (represented by w and b) and the dual form (represented by α_i). Equations 4 and 5 are sufficient and necessary conditions for a maximum of equations because they are convex and KKT. Equation (5) is differentiated partially in terms of saddle points (w_0, b_i, α_0).

$$\partial L \quad \partial w_0 = 0$$
$$\text{i.e } w_0 \quad = \sum_{i=1} \alpha_i y_i x_i$$

And $\partial \mathrm{L}/\,\partial \mathrm{b}_0 = 0$

i.e., $\sum_i^1 \alpha_i y_i = 0$

Alter equation (6) and equation (7) in equation (5). The dual form derives from the primary form.

$$L_d(\alpha) = \sum\alpha_i - 1/2\sum_{i=1}^{1}\alpha_i\alpha_j y_i y x_i^T x_j$$

To find the optimal hyperplane, a dual lagrangian (Ld) must be maximised with respect to nonnegative α_i (i.e., α_i must be in the nonnegative quadrant) and equality constraints.

$$a_i \geq 0\,, i = 1,2 \ldots \ldots 1$$
$$\sum_{i=1}\alpha_i y_i = 0$$

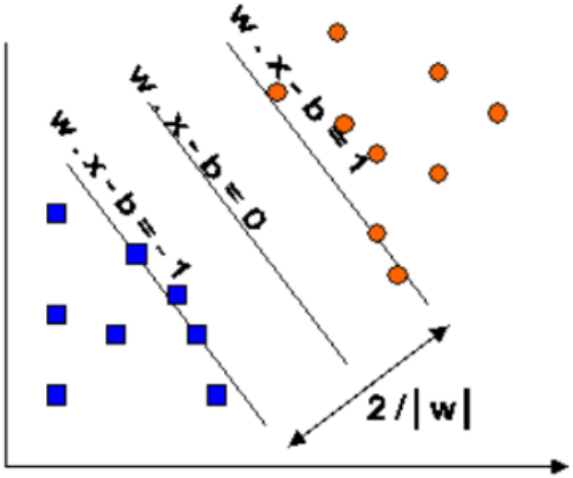

Fig 3. Maximum Margin Hyperplanes for a SVM trained with Samples from Two Classes

4.1.1 KERNEL SELECTION OF SVM

The Φ function transforms x_i into a higher-dimensional space (possibly infinite dimensions). In this higher-dimensional space, the SVM identifies the hyperplane with the greatest margin for linear separation. $C > 0$ is the error term penalisation parameter. The alternate name of the kernel function is $K(xi, xj) \equiv \Phi(xi)T\ \Phi(xj)$ [17]. The selection of the most appropriate kernel function from the numerous SVM options is a further research obstacle. On the other hand, the two kernel functions listed below [17] are especially useful for more widespread applications:

- Linear kernel: $K(x_i, x_j) = x_i^T x_j$.
- Polynomial kernel:

$$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d,\ \gamma > 0$$

- RBF kernel:

$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right),\ \gamma > 0$$

- Sigmoid kernel:

$$K(x_i, x_j) = \tanh\left(\gamma x_i{}^T x_j + r\right)$$

Here γ, r and d are kernel parameters.

RBF is the primary kernel function in these widely used kernel functions for the following reasons [2]:

In contrast to a linear kernel, the RBF kernel maps samples into a higher-dimensional space in a nonlinear fashion. The RBF kernel has fewer hyperparameters than its polynomial counterpart. RBF kernel presents fewer numerical challenges.

4.1.2 Utilization of SVM for Model Selection

Without model selection, an SVM analysis is incomplete. Recent studies have demonstrated that SVM outperforms other data classification techniques. A number of parameters whose values influence the generalisation error must be fine-tuned for the operation to be successful. Model selection denotes to the fine-tuning process the parameters of a model. The only parameter that must be adjusted when employing the linear support vector machine is the cost parameter C. However, linear SVM is frequently used to solve problems that can be separated

linearly. There are numerous things that are challenging to distinguish. For example, data from the Space Shuttle and data from satellites cannot be separated in a linear fashion. When solving classification problems [4 & 5], we frequently use nonlinear kernels to obtain the value of cost parameter (C) and the kernel parameters (γ, d). When determining the optimal parameter set for cross validation, the grid-search method is frequently employed. Upon applying this parameter set to the training dataset, the classifier can then be obtained. Use the classifier to categorise the testing dataset so that you can evaluate the accuracy of the generalisation. There are numerous things that are challenging to distinguish. For example, data from the Space Shuttle and data from satellites cannot be separated in a linear fashion. When solving classification problems, we frequently use nonlinear kernels to determine the value of the cost parameter (C) and the kernel parameters (γ, d). When determining the optimal parameter set for cross validation, the grid-search method is frequently employed. Upon applying this parameter set to the training dataset, the classifier can then be obtained. Use the classifier to categorise the testing dataset so that you can evaluate the accuracy of the generalisation.

4.2 CNN

Convolutional Neural Networks, also referred to as CNNs or ConvNets, are a type of multi-layer neural network that is inspired by the optical systems of living organisms. According to Hubel and Wiesel, animal cells in the small receptive field of the visual cortex are responsible for light detection. [4.] In 1980, Kunihiko Fukushima introduced the neocognitron [5], a multilayered neural network capable of learning to recognise hierarchical visual patterns, to the world. This network served as the model for CNN's design. LeCun et al. created the practical CNN model [6] [7] and LeNet-5 [8] in 1990. After being trained with the backpropagation [9] algorithm, LeNet-5 was able to recognise visual patterns from unprocessed pixels without a separate mechanism for feature engineering. This was made possible by the network's ability to recognise visual patterns. In addition, conventional feedforward neural networks with comparable network sizes had more connections and parameters, whereas CNN models had fewer connections and parameters, making model training easier. CNN's performance in complex problems, such as the classification of high-resolution images, was hindered at the time by a lack of large training data, an improved regularisation method, and insufficient computing power. ImageNet [10], LabelMe [11], and other sources provide access to more extensive datasets with millions of high-resolution, labelled data spanning thousands of categories. Since the introduction of powerful GPU machines and a refined regularisation

method, CNN's performance in image classification tasks has significantly increased. This advancement was made possible by the combination of these two innovations: by imposing a local connectivity pattern between neurons in adjacent layers, CNNs take advantage of their spatially local correlation. As shown in the diagram, layer m neurons are connected to a local subset of layer m-1 neurons, and layer m-1 neurons have contiguous receptive fields (2a)

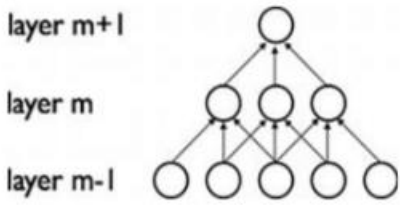

Fig 4(a): Graphical Flow of Layers showing Connection between Layers

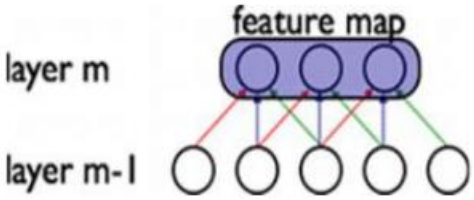

Fig 4(b): Graphical Flow of Layers showing sharing of weights

Each sparse filter is replicated across the entire visual field when the CNN algorithm is utilised. Using the same weight vector and bias, these units then construct the feature map. Figure 4b depicts three concealed units on the same feature map as Figure 4a. Since the weights of each colour are shared, it is essential that they are identical.

By adding the gradients of the shared parameters, it is possible to calculate the gradient of the shared weights. Due to this replication, features can be identified regardless of their position within the field of vision. Additionally, weight sharing reduces the total number of available free learning parameters. This control enables CNN to achieve a more accurate generalisation of vision problems. CNN uses a nonlinear downsampling technique known as max-pooling in its reporting. This method divides the input image into non-intersecting rectangles. The production of each subregion reaches its maximum level.

4.2.1 Convolution layer:

CNN's network consists of multiple layers, the first of which is the convolution layer. This diagram illustrates the composition of this layer (3). Also included are a bias term, a function expression, and a convolution mask. Together, they produce the layer's final product. In the image below, a 32 x 32 input feature map with a 5 x 5 convolution mask has been applied. This operation will result in a 28-by-28 matrix. The sigmoid function is then applied to the matrix [7] following the application of the bias.

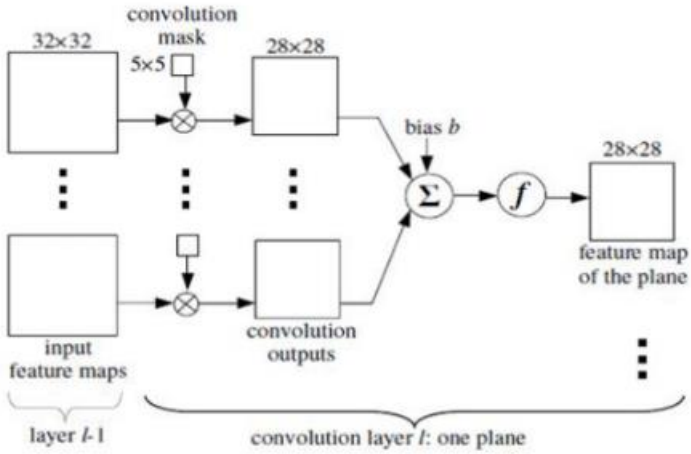

Fig 5. Convolutional Layer Working

4.2.2 Sub sampling layer

The subsampling layer follows the convolutional layer immediately. The number of planes in this layer is identical to that of the convolutional layer. This layer is intended to compress the feature map in order to conserve space. After dividing the image into 2x2 blocks, averaging is performed on the blocks. The subsampling layer remembers only the relative nature of the relationships between features.

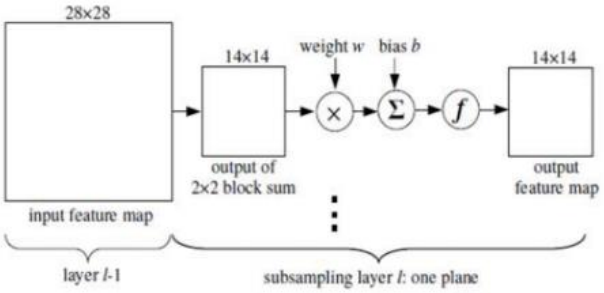

Fig 6. Sub-Sampling Layer Working

A layer with complete connectivity (FC Layer) As illustrated in Figure 7, the CNN conclusion is composed of interconnected layers. This layer receives input from all of the neurons in the layer below it and generates output using the neurons in the layer above it.

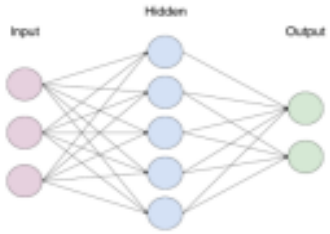

Fig 7. Fully Connected Layer

4.3 LSTM

Due to its superior ability to model and learn from sequential data, long short-term memory, also known as LSTM, has recently gained popularity among natural language processing researchers (NLP). On a variety of publicly available benchmarks, including sentence classification [9] [10], language modelling, and sequence-to-sequence predictions [6], this algorithms have demonstrated state-of-the-art performance. The LSTM algorithm solves the RNN gradient vanishing and exploding problem. LSTM architecture substitutes the hidden vectors in recurrent neural networks with memory blocks containing gates. It is a proof that, to be useful in achieving advanced solutions for a variety of problems, including speech recognition [3], and by practising appropriate gating weights, it can theoretically maintain long-term memory. Note: Hochreiter and Schmidhuber (1997) proposed LSTM as a solution to the issue of long-term learning dependence. The LSTM is responsible for storing unique memory cells that are only updated and displayed when necessary [3]. Three levels comprise the LSTM gates mechanism: (1) the input gate, (2) the forget gate, and (3) the output gate . As depicted in Figure 2, every LSTM unit has a memory cell, and ct denotes the values that represent the state at any given time. The sigmoid gate controls both reading and modification in addition to the input gate it, the forget gate ft, and the output gate ot. The following is the formula for the

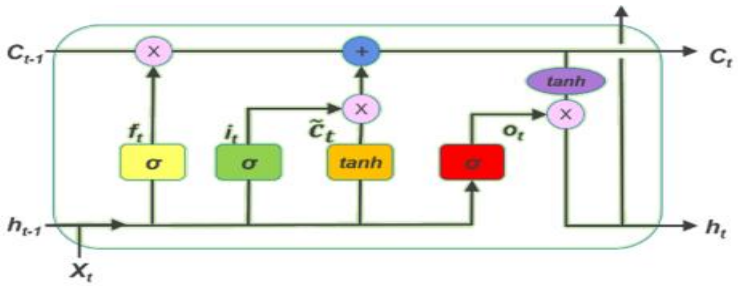

Fig 8. LSTM Architecture

Currently, the model obtains its information from two distinct external sources (ht-1 and xt). When calculating the node states of the hidden layer, keep in mind that the input gate, output gate, forget gate, and xt all influence the node state simultaneously.

A step-by-step explanation of the LSTM cell and its gates is provided below:

1 Input Gate:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (1)$$

$$\check{C}_t = \tanh\,(W_C \cdot [h_{t-1}, x_t] + b_C \quad (2)$$

2 Forget Gate:

$$f_t = \sigma\big(W_f \cdot \big[h_{t-1}, x_t + b_f\big) \quad (3)$$

3 Memory State:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

4 Output Gate:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t * \tanh\,(C_t) \quad (6)$$

Evaluation The multi-label evaluation steps of the confusion matrix in the following equations:

$$\text{Acc} = \frac{\sum_{i=1}^{l} \frac{TP_i + TN_i}{TP_i + FN_i + TN_i + FP_i}}{l} * 100\% \quad (7)$$

$$\text{Presisi} = \frac{\sum_{i=1}^{l} TP_i}{\sum_{i=1}^{l} (FP_i + TP_i)} * 100\% \quad (8)$$

$$\text{Recall} = \frac{\sum_{i=1}^{l} TP_i}{\sum_{i=1}^{l} (TP_i + FN_i)} * 100\% \quad (9)$$

$$\text{F1 score} = \frac{2 * \text{Presisi} * \text{Recall}}{\text{Presisi} + \text{Recall}} \quad (10)$$

4.3.1 Optimization

SGD, Adam, RMSProp, and others are deep learning optimizers. This research trained the data with Adam and RMSProp. Adam Optimizer controls sparse gradients [4]. It's an expansion of stochastic gradient descent, which is used in deep learning and NLP.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (11)$$

$$v_t = \beta_2 v_{t-1} + (1 + \beta_2) g_t^2 \text{Type equation here.} \quad (12)$$

g is the minibatch gradient, and m and v are its first two moments. RMSProp lets users customise each parameter's learning rate. It divides the weight learning rate by a moving average of recent gradient magnitudes [35].

$$v(w,t) := \gamma v(w,t-1) + (1-\gamma)(\nabla Q_i(w))^2 \quad (13)$$

γ ➔ forgetting factor. As well as the parameters are updated as,

$$w := w - \frac{n}{\sqrt{v(w,t)}} \nabla Q_i(w) \quad (14)$$

4.3.2 Pre-Processing

4.3.2.1 One-hot Encoding

A one-hot encoding is performed as the very first step in the pre-processing phase of this research. When categorical text data needs to be converted to numerical form, one-hot encoding is the method of choice. At the moment, algorithms for machine learning are unable to directly process data that is categorical. Converting categorical data to numerical data is required. Deep learning methods, such as long-term memory recurrent neural networks, are utilised in research pertaining to sequence classification.

The process of dividing a text into tokens, which may be words, phrases, symbols, or other meaningful components, is known as tokenization. Tokens can take on a variety of forms. There is a wide range of sizes and contours to choose from when purchasing tokens. The process of separating a text into individual tokens that are incomprehensible is known as tokenization. This is necessary in order for any type of processing to take place. This procedure tokenizes each sentence individually after the text has been broken up into sentences. Therefore, punctuation marks have to go in their very own designated spaces within the text. The generation of training data is one of the characteristics that results from tokenization. Padding is utilised during the training process of the decoder on a sentence-by-sentence basis to ensure that an accurate determination of the end of each sentence can be made during the training phase.

4.4 CNN-LSTM Network

There is a theory that states that CNNs are straightforward computational models of the mammalian visual cortex [17]. CNNs are a form of biologically motivated feed-forward ANN. CNNs draw their inspiration from the processes that occur in living organisms. 2D and 3D convolutional neural networks are typically utilised in the processing of images and videos, whereas 1D convolutional neural networks are typically utilised in the recognition of audio and text (as a time series data). 1D-CNNs are particularly effective when applied to problems that require the recognition and forecasting of time series. Recent applications of networks include early diagnosis, monitoring of structural health, identification, and detection and identification of anomalies [18]. An additional illustration is the detection of early anomalies. Because our data is a time series of vibration signals, we will be using one-dimensional convolutional neural networks. The output of a convolutional layer that is located in the ith layer (vij x) at the position x of the jth feature map is represented by the following expression.

$$v_{ij}^{x} = g\left(b_{ij} + \sum_{m}\sum_{p=0}^{P_i-1} w_{im}^{p} v_{(i-1)m}^{x-p}\right)$$

Where m is the index of the feature map from the layer below the current layer that is connected to the feature map from the layer above it. n is the total number of feature maps that are being taken into consideration, and Wim p is the weight of position p in the mth feature map. Pi stands for the kernel width of the spectral dimension, bij for the bias of the ith layer's jth feature map, and g for the activation function. All three are denoted by the superscripts pi, bij, and g. After one or more CNN layers have been added, a Pooling layer is typically added to provide invariance by reducing the resolution of the feature map [20]. The final layer of CNN models is referred to as the pooling layer. There is a connection between each successive layer of pooling and the one that came before it. The max-pooling operation is the one that is used the vast majority of the time:

$$\bar{u}_a = \max_{1\le j<k}\left(u_n^j\right)$$

In this equation, " $u_n{}^j$ " stands for the jth element of the nth patch, " $\bar{u_n}$ " stands for the maximum pooling sample of the n_{th} patch, and "k" stands for the size of the patch. [21] Long-term memory networks, also known as LSTM networks, are neural networks that regulate access to memory cells through the use of a specialised gating mechanism. LSTM networks are able to maintain signals for significantly longer durations than conventional recurrent neural networks can due to the gates' ability to prevent the rest of the network from altering the contents of the memory cells for multiple time steps. In addition, this makes it possible for

errors to spread over an extended period of time. The LSTM was developed by Hochreiter and his colleagues [21] in order to model temporal sequences and the long-range dependencies between them with greater precision than is possible with conventional RNNs. Each LSTM block consists of three gates: an input gate, a forget gate, and an output gate. These gates are referred to collectively as "gates." Sigmoid activation functions are responsible for the generation of numbers in the range of zero to one, and gates are a method for selectively transmitting information.

$$\sigma(t) = \frac{1}{1 + e^{-1}}$$

The gate will not let anything that has a value of zero through, but anything that has a value of one will be allowed to proceed. The following equations are what describe the gates mathematically: [22]

$$i_1 = \sigma(w_i[h_{1-1}, x_1] + b_i)$$
$$f_1 = \sigma\big(w_f[h_{t-1}, x_t] + b_f\big)$$
$$o_t = \sigma(w_a[h_{t-1}, x_t] + b_e)$$

Where it is the input gate, ft is the forget gate, ot is the output gate, sigmoid is the activation function, wx is the weight of the respective gate (x), ht1 is the output of the previous LSTM block at time-step t-1, xt is the current input, and bx is the biases of the respective gate. bx is the biases of the respective gate. It is the input gate, the forget gate is denoted by ft, and the output gate is denoted by ot (x). You can determine the cell state vector, denoted by ct, as well as the LSTM unit output vector, denoted by ht, by using the following formulas:

$$\tilde{c}_1 = \tanh\,(w_c[h_{t-1}, x_1] + b_c)$$
$$c_t = f_t * c_{1-1} + i_1 * z_1$$
$$h_t = o_t * \tanh\,(c_1)$$

Where $\tilde{c}t$ represents the candidate vector for cell state vector (how much we decide to update each state vector); tanh is the hyperbolic tangent activation function and $(*)$ denotes the Hadamard product. An illustration of an LSTM block is provided in the following diagram.

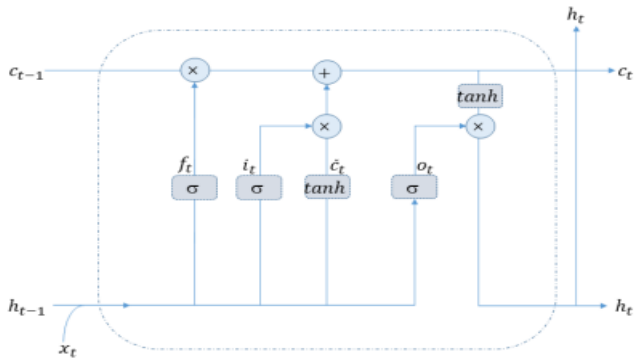

Fig 9. LSTM BLOCK

CNN-LSTMs are useful for a wide range of applications, including visual learning, speech recognition, and the processing of natural languages [8]. Both convolutional neural networks (CNNs) and long short-term memory (LSTMs) have been shown to be equally effective at predicting temporal sequences [9]. Both the precision and accuracy of predictions are improved by the CNN-LSTM network [10]. When working with large data sets or complex temporal sequence problems, this is an especially useful skill to have. We will be able to explain the temporal sequence prediction if we are able to observe a dynamical system that spans a temporal region that is represented by an MN grid that has M rows and N columns. A time-varying measurement of the parameter P is stored in each grid cell of the matrix. As a consequence of this, the number of features can be accurately represented by a tensor that has the dimensions P, M, and N. We are able to divide the dataset into samples of the same length and obtain a series of tensors that are denoted by the numerals $\hat{X}_1$, $\hat{X}_2$, $\hat{X}_3$…, $\hat{X}_n$, if the features are recorded at regular intervals throughout the data collection process. The goal of the temporal sequence prediction problem is to maximise the following conditional probability in order to determine the most likely kth sequence of observations given the jth observation. This will be accomplished by predicting the most likely outcome of the problem.

$$\hat{Y}_{t+1}, \dots, \hat{Y}_{t+k} = \underset{X_{t+1}, \dots X_{t+1}}{\arg\max} p\left(X_{t+1}, \dots, X_{t+k} \mid \hat{X}_{t-j+1}, \hat{X}_{t-j+2}, \dots, \hat{X}_t\right)$$

4.5 Decision Tree Classifier:

The decision tree algorithm is a data mining induction technique that divides a set of records into classes repeatedly using either a depth-first greedy approach or a breadth-first approach until all of the data items belong to that class. This process can be carried out using either a depth-first approach or a breadth-first approach. The starting point of a decision tree is called the root node, and it is made up of the nodes that come before and after it. Unknown data

records are categorised into subcategories by using the tree structure. In order to determine the most effective method of splitting at each node within the tree, measures of impurity are used. The leaves of the tree are made up of the class labels that were utilised in the process of grouping the data items. The first step of the decision tree classification method is to construct the tree, and the second step is to prune it. The crown of a tree is the first layer to form. At this point, the tree will be divided multiple times until all of the data items have the same class label. This process will continue until the tree is complete. It is a very challenging task that requires a lot of processing power due to the fact that the training data set is gone through multiple times. When trees are pruned, work is done from the ground up. Its purpose is to reduce the likelihood of the algorithm's predictions and classifications being incorrect by acting as a check against overfitting (noise or much detail in the training data set). Overfitting data can result in misclassification errors when using decision tree algorithms. Pruning a tree requires less effort than growing a tree does due to the fact that the training data set only needs to be scanned once. The decision tree that the proposed system uses provides the end user with an improved method for classifying tweets into positive and negative categories. It is done by comparing the items with the highest frequency in the training data with the items with the highest frequency in the test data. This makes it simple to organise the data into categories. The results of the proposed system were compared to those of other classification methods and research papers that had previously used the Twitter dataset. This allowed the researchers to determine how well the proposed system works. The proposed system's efficiency was evaluated based on a number of different metrics, including its precision, recall, classification accuracy, and f-measure.

4.6 Random Forest

Learning algorithms known as ensemble classification methods create a group of classifiers as opposed to just one, vote on how new data points should be categorised, and then classify the data based on the results of that vote. Bagging, Boosting, and Random Forest are the three ensemble classifiers that are used the most frequently [28] (RF). The random forest algorithm is a form of supervised machine learning that utilises group learning as its primary data processing method. The goal of the learning strategy known as ensemble learning is to produce a more accurate prediction model by combining several distinct implementations of the same algorithm. The algorithm known as "random forest" gets its name from the fact that it combines multiple algorithms that are all of the same kind—for example, multiple decision trees—in

order to create a forest of trees. Both regression and classification problems are amenable to being solved with the assistance of the random forest algorithm. A group of classifiers organised in a tree-like fashion is referred to as an RF classifier. It's a more difficult variation of Bagging that involves the element of chance [28]. In RF, rather than using the best split for all variables, the best split is used for a subset of predictors that are chosen at random at each node. This allows for more accurate results. Alterations are made to the primary data set before a new training data set can be created. Then, a tree is created by selecting its characteristics at random and growing it. It is not necessary to prune trees that have reached their mature size [27, 28]. This strategy gives RF an accuracy that can't be beat. RF is also quick, doesn't fit too well, and can make as many trees as the user wants [27, 28]. For both classification and regression, this is how the random forests algorithm works: Make bootstrap samples from the original data. Grow an unpruned classification or regression tree for each of the bootstrap samples, with the following change: at each node, instead of choosing the best split among all of the predictors, randomly sample a subset of the predictors and choose the best split among those variables. (You can think of bagging as a special case of random forests where the number of predictors is equal to p.) By putting together what the trees say, you can predict new data (i.e., majority votes for classification, the average for regression) (i.e., majority votes for classification, the average for regression).

4.7 XGBOOST:

Gradient-boosted decision trees are used in XGBoost, which is a fast and efficient way to do this. XGBoost stands for eXtreme Gradient Boosting. Most predictive models aren't as good as XGBoost. Because of this, we will use it to put our tweets into groups.

XGBoost is one of a group of algorithms called "boosters" that help weak learners get better. A slow learner is someone who isn't much better than someone who just guesses at things. Boosting is a process where trees are grown one after the other using information from a tree that has already been grown. Over time, this process slowly learns from the data and tries to get better at making predictions. Gradient-boosted decision trees are used in XGboost, which is a fast and efficient way to do this. At its core, it is based on a framework called "gradient boosting." This machine learning algorithm has the following parts:

4.7.1 Parallel Computing

It is set up for parallel processing, which means that when you run XG boost, it will automatically use all of your computer's cores.

- Regularization: Regularization is a way to avoid overfitting in linear and tree-based models.
- Missing Values: XGBoost is built to handle missing values on the inside. The missing values are dealt with in a way that lets the model pick up on any trend in missing values.In addition to regression, classification, and ranking problems, user-defined objective functions can also be used. Given a set of parameters, an objective function is used to judge how well the model works. It also lets users set up their own metrics for judging.
- Save and Load: With XGBoost, we can save and load our data matrix and model. If we have a lot of data, we don't have to redo the calculations. Instead, we can just save the model and use it again in the future.

Unlike GBM, which stops pruning when a negative loss is found, XGBoost grows the tree to its maximum depth and then prunes backward until the improvement in the loss function is below a threshold.

4.8 Naïve Bayes:

Assigning the class $c^* = \arg\max_c P(c \mid d)$ to a given document d is one method of text categorization. The Naive Bayes (NB) classifier is derived by first recognizing that, accordingto Bayes' rule,

$$P(c \mid d) = \frac{P(c)P(d \mid c)}{P(d)}$$

$P(d)$ has no bearing on the selection of c^*. Naive Bayes decomposes the term $\underline{P}(d \mid c)$, by

$$P_{\mathrm{NB}}(c \mid d) := \frac{P(c)\left(\prod_{i=1}^{m} P(f_i \mid c)^{n_i(d)}\right)}{P(d)}$$

Our training technique uses add-one smoothing to estimate the relative frequency of $P(c)$ as well as $P(fi \mid c)$. In spite of it being simple as well as the fact that its conditional independence assumption doesn't hold in the real-world situations, Naive Bayes-based text categorization performs admirably (Lewis, 1998); indeed, Domingos and Pazzani (1997) demonstrate that Naive Bayes is optimal for certain problem classes with highly dependent features.

CHAPTER 5

DATASET PREPARATION &ANALYSIS OF PROPOSED METHODS

5.1 DATASET PREPARATION

The EEG readings of over 80 patients, each of whom took part in multiple trials, make up the bulk of the data that we have collected for this project from Kaggle.

The capacity to lessen or get rid of the sensory consequences of one's actions is present in the brains of many different animals, including the human brain. This is accomplished through the utilisation of a corollary discharge forward model system by the nervous system. During this stage of the process, the sensory cortex receives what the motor cortex refers to as a "reference copy" of an upcoming motor plan. The information contained in the "efference copy" is used by the sensory cortex to generate a "corollary discharge" representation of the anticipated sensory consequences of the upcoming motor act. This representation is based on the information. If you move your eyes from left to right, for example, your brain realises that the environment has not altered but simply follows along with the new visual input. When you speak, the auditory cortex in your brain reacts in a manner that is distinct from how it normally does when it hears the sound of your own voice.

Schizophrenia is a persistent mental illness that impacts approximately 1 percent of the world's population. It may be challenging for patients with schizophrenia to differentiate between internally generated and externally generated stimuli. This difficulty may be caused by one or more problems with the corollary discharge process of the nervous system. This is one of the possible explanations for some of the symptoms of schizophrenia. Therefore, studying this process and how it relates to disease symptoms could help us gain a better understanding of the abnormal brain processes that are present in patients who have this disease. Patients who have Alzheimer's disease have been shown to have these abnormal brain processes.

The authors of a previously published EEG study [14] utilised a straightforward button-pressing task in order to investigate the corollary discharge in both participants who had schizophrenia and healthy controls. During this activity, participants were given the option to either (1) immediately press a button to generate a tone, (2) listen passively to the same tone, or (3) press a button without generating a tone. Participants in the study were selected from a pool of people who had a history of receiving a diagnosis of schizophrenia. Patients with

schizophrenia were unable to suppress the N100, which is a negative deflection in the EEG brain wave that occurs one hundred milliseconds after the beginning of a tone. This occurred when the patients pressed a button to produce the tone. When patients were instructed to simply listen to sounds without actively participating, the N100 was suppressed. The number of people in the sample was increased so that the results of the earlier study could be repeated. In this study, the EEG data of 10 healthy individuals and 13 patients with schizophrenia from a previous study were combined with the EEG data of 22 healthy individuals and 36 patients with schizophrenia.

5.2 Pre-Processing

The EEG signals from the dataset go through a series of pre-processing steps to prepare them for analysis. In order to generate a total of 7092 temporal samples without any overlap, each of the 19 recorded EEG signals was first segmented into 25-second frames. As a direct consequence of this, each frame of an EEG signal has the dimensions of 709219. Following that, the subsequent EEG frames were subjected to the z-score and L2 methods of normalisation, respectively. Traditional ML and DL models benefit from having their precision and overall performance improved by normalising the EEG signals. The dataset has been obtained from Kaggle and following steps describe the pre-processing part of the dataset.

Preprocessing Steps :

[1] Making average n rows in a matrix and also reshaping with mean.
[2] Using electrodes list and taking average creating array for dependent and independent variable.
[3] Using data button-tone data, column label dividing them with number of person.
[4] Using number of trails of process (9216.0)
[5] Calculating total trails with appropriate number of measurements using dependent and independent variable counter functions.
[6] Splitting data into train and test then normalizing and setting (norm = max) and then reshaping
[7] Again splitting data with shuffling setting True.
[8] Finally, reshaping the independent data using normalizing data with length of electrodes data in both data (test and train).

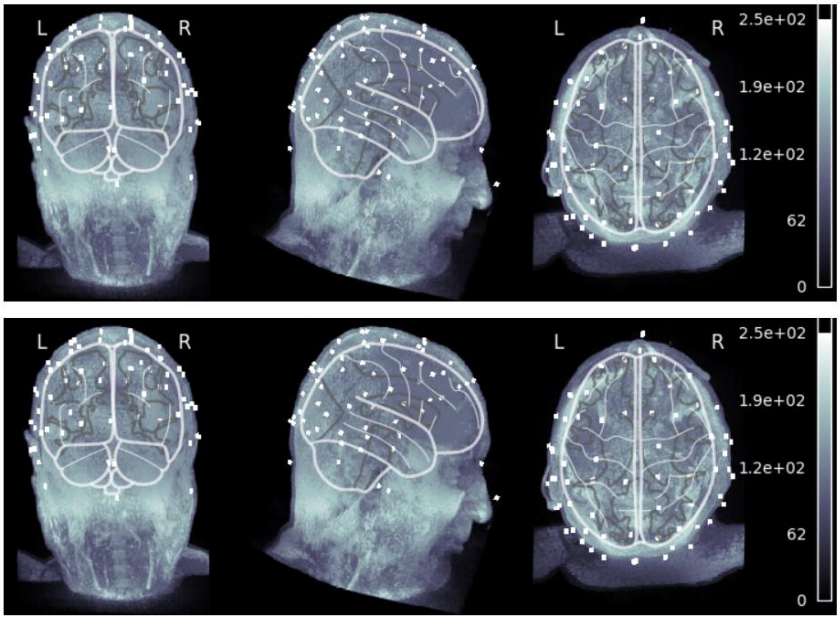

Fig 10: Brain stumuli of patients during EEG

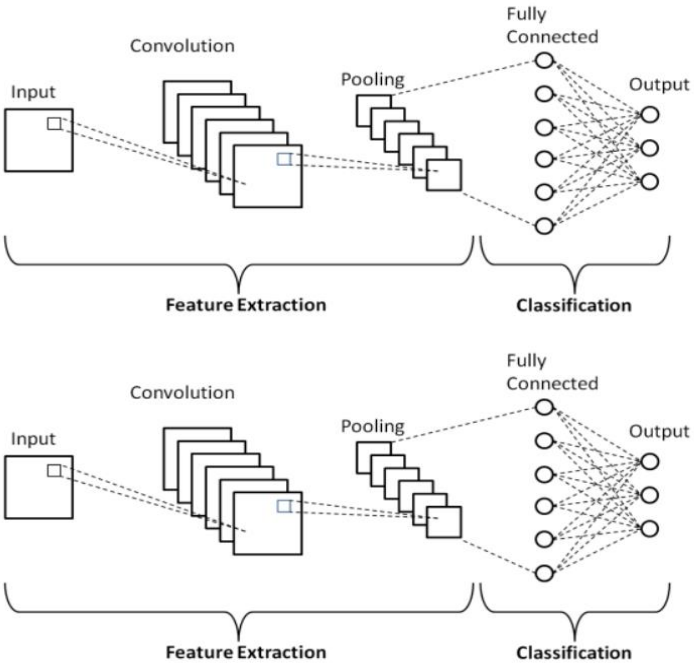

Fig 11: Modelling Methodology

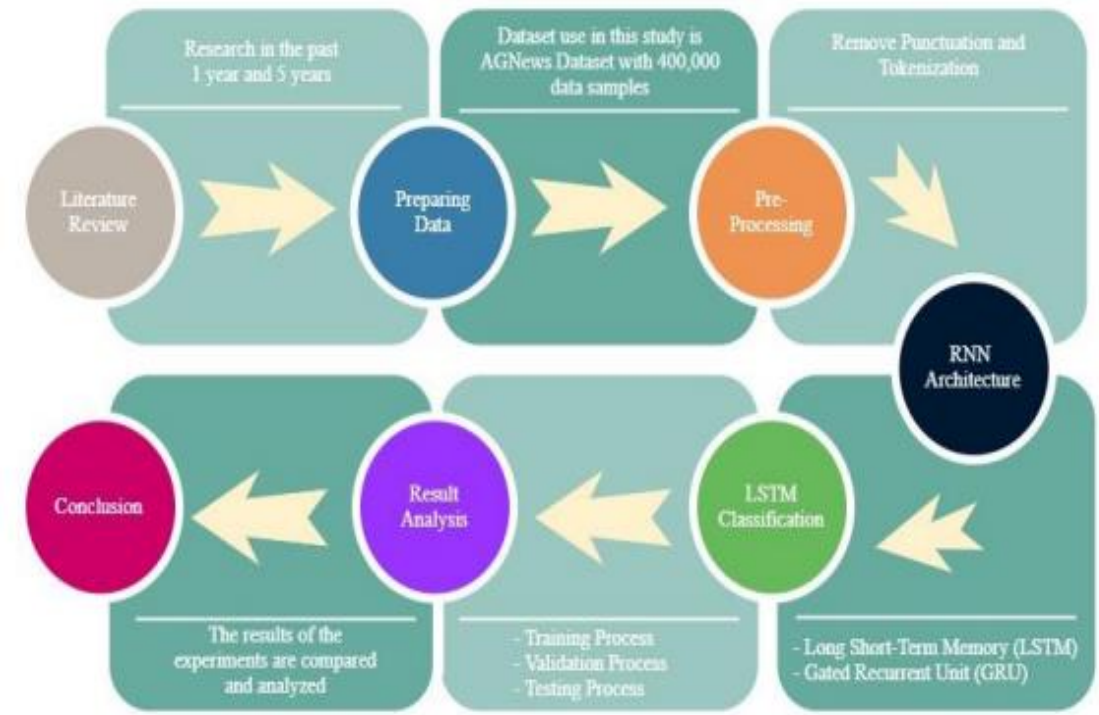

Fig 12: Research Methodology

5.3 Data leakage, Pipeline and Hyper Parameter tuning

Data leakage is a serious concern when utilising machine learning to build predictive models. When information from outside the training dataset is utilised to generate the model, this is known as data leaking. Predictive modeling's purpose is to create a model that can make accurate predictions based on fresh data not observed during training. This is a complicated situation. It's troublesome since we can't test the model against something we don't have. As a result, we must estimate the model's performance on unknown data by training and assessing it on a portion of the data. This approach underpins cross validation and other advanced processes that aim to limit the variation in this estimate. Forecasting models may be unduly optimistic, if not entirely wrong, as a result of data breaches. Data leakage occurs when information from outside the training dataset is used to create the model. This additional input may allow the model to learn or know something it wouldn't have known otherwise, undermining the mode's expected performance. Data preparation (for example, dealing with missing information, scaling/encoding, and feature extraction) is the first step in the machine learning process. While learning this method, we do the data preparation one step at a time. This may take some time since we need to prepare both training and testing data. Pipelines allow us to streamline this process by combining the preliminary processes and making model

modification and monitoring more convenient. Scikit-Pipeline Learn's class provides a framework for conducting a series of data changes followed by an estimation (Mayo, 2017). There are many benefits when implementing a Pipeline:

- **Convenience and encapsulation:** We call fit as well as make the prediction only once about data to fit an entire sequence of estimators.
- **Joint parameter selection:** We can do a grid-search over the parameters of all the estimators in pipeline.
- **Cross-Validation:** Pipelines are responsible in helping to avoid any data leakage from testing data into trained model during the cross-validation. It's achieved by making sure that same samples have been used for training the transformers as well as predictors.

5.4 Performance Analysis of Proposed Methods

5.4.1 Machine Learning Algorithms

Research Studies related to the study of EEG signals data for Schizophrenia patients using machine learning techniques have been published. We have trained our model on Logistic Regression, Decision Tree, Classifier, Random Forest classifier, Stochastic Gradient Descent, Gradient Boosting Classifier, XGB Classifier, Bernoulli Naïve Bayes Classifier.

The Accuracies for all the models are presented in the table below:

Algorithm	Accuracy (Test)
Logistic Regression	60.74
Decision Tree Classifier	57.99
Random Forest Classifier	65.96
Stochastic Gradient Descent	59.41
Gradient Boosting Classifier	59.27
XGB Classifier	69.27

Bernoulli Naïve Bayes Classifier	53.14

Table 1: Accuracy of Machine Learning Models

Since the above accuracies aren't good enough for us, we will in order to reduce the data leakage and get better accuracy use a Pipeline to minimize the error rate or data leakage which help to boost our accuracy as we tried seven different classifiers. Logistic Regression, Decision Tree, Classifier, Random Forest classifier, Stochastic Gradient Descent, Gradient Boosting Classifier, XGB Classifier, Bernoulli Naïve Bayes Classifier. Pipelines contain our pre-processing procedures and models, simplifying the machine learning workflow. If necessary, we may perform several pre-processing steps before fitting a model into the pipeline.

After using Pipeline to reduce the data leakage our accuracy for the above mentioned algorithms are presented below in the form of Classification Report and Confusion Matrix.

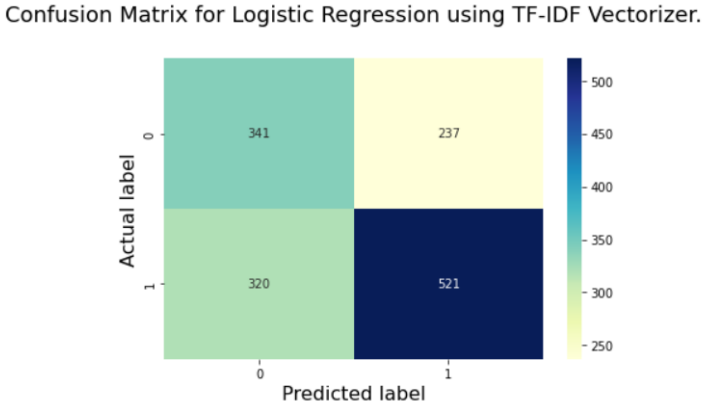

Fig 13: Confusion Matrix for Logistic Regression (Pipeline Version)

```
Classification Report for Logistic Regression is :
              precision    recall  f1-score   support

         0.0       0.52      0.59      0.55       578
         1.0       0.69      0.62      0.65       841

    accuracy                           0.61      1419
   macro avg       0.60      0.60      0.60      1419
weighted avg       0.62      0.61      0.61      1419
```

Fig 14: Classification Report for Logistic Regression (Pipeline Version)

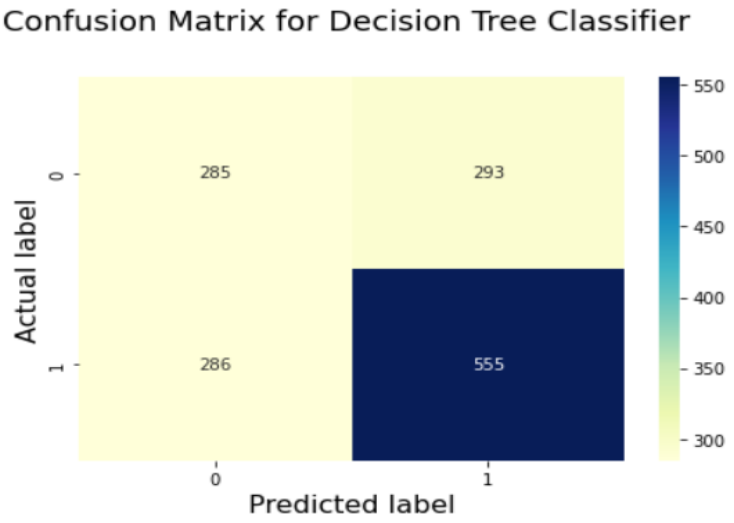

Fig 15: Confusion Matrix for Decision Tree Classfier (Pipeline Version)

Classification Report for Decision Tree Classifier is :

	precision	recall	f1-score	support
0.0	0.50	0.49	0.50	578
1.0	0.65	0.66	0.66	841
accuracy			0.59	1419
macro avg	0.58	0.58	0.58	1419
weighted avg	0.59	0.59	0.59	1419

Fig 16: Classification Report for Decision Tree Classifier (Pipeline Version)

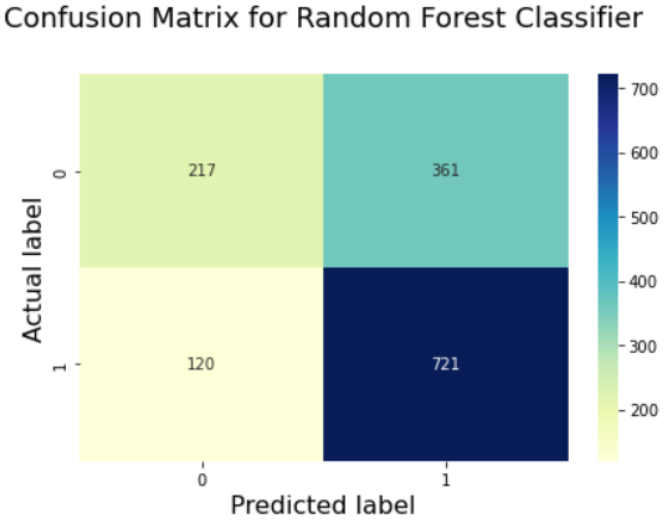

Fig 17: Confusion Matrix for Random Forest Classifier

Classification Report for Random Forest Classifier is :

	precision	recall	f1-score	support
0.0	0.64	0.38	0.47	578
1.0	0.67	0.86	0.75	841
accuracy			0.66	1419
macro avg	0.66	0.62	0.61	1419
weighted avg	0.66	0.66	0.64	1419

Fig 18: Classification Report for Random Forest Classifier

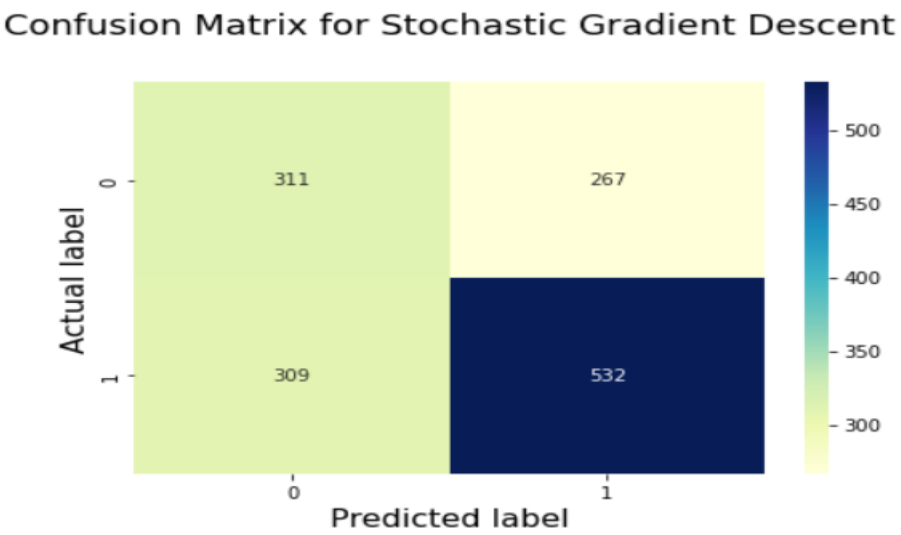

Fig 19: Confusion Matrix for Stochastic Gradient Descent

Classification Report for Stochastic Gradient Descent is :

	precision	recall	f1-score	support
0.0	0.50	0.54	0.52	578
1.0	0.67	0.63	0.65	841
accuracy			0.59	1419
macro avg	0.58	0.59	0.58	1419
weighted avg	0.60	0.59	0.60	1419

Fig 20: Classification Report for Stochastic Gradient Descent

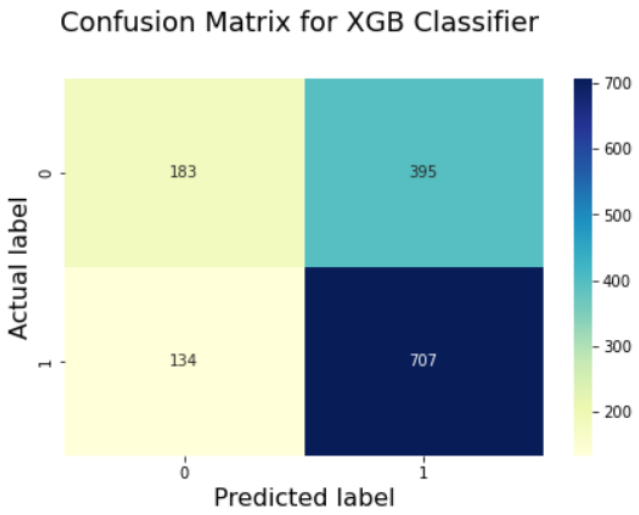

Fig 21: Confusion Matrix for Gradient Boosting Classfier

Classification Report for Gradient Boosting Classifier is :

	precision	recall	f1-score	support
0.0	0.00	0.00	0.00	578
1.0	0.59	1.00	0.74	841
accuracy			0.59	1419
macro avg	0.30	0.50	0.37	1419
weighted avg	0.35	0.59	0.44	1419

Fig 22. Classification Report for Gradient Boosting Classifier

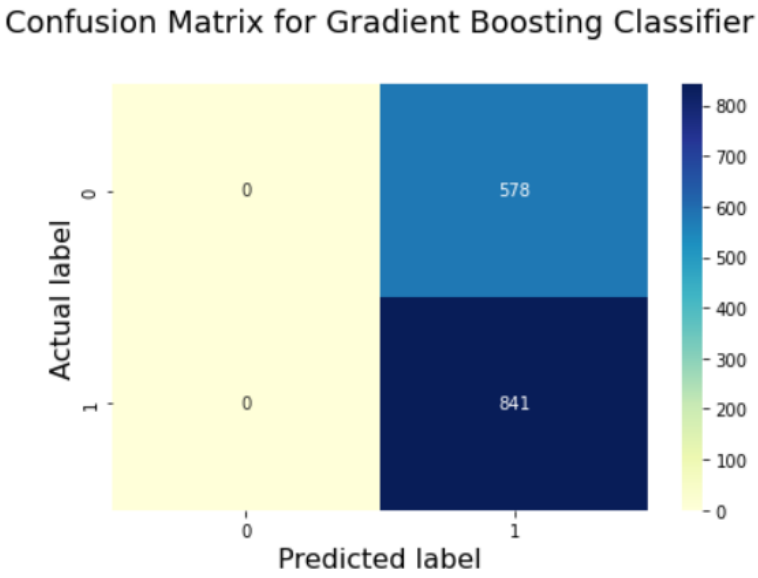

Fig 23: Confusion Matrix for XGB Classifier

Classification Report for XGB Classifier is :

	precision	recall	f1-score	support
0.0	0.58	0.32	0.41	578
1.0	0.64	0.84	0.73	841
accuracy			0.63	1419
macro avg	0.61	0.58	0.57	1419
weighted avg	0.62	0.63	0.60	1419

Fig 24: Classification Report for XGB Classifier

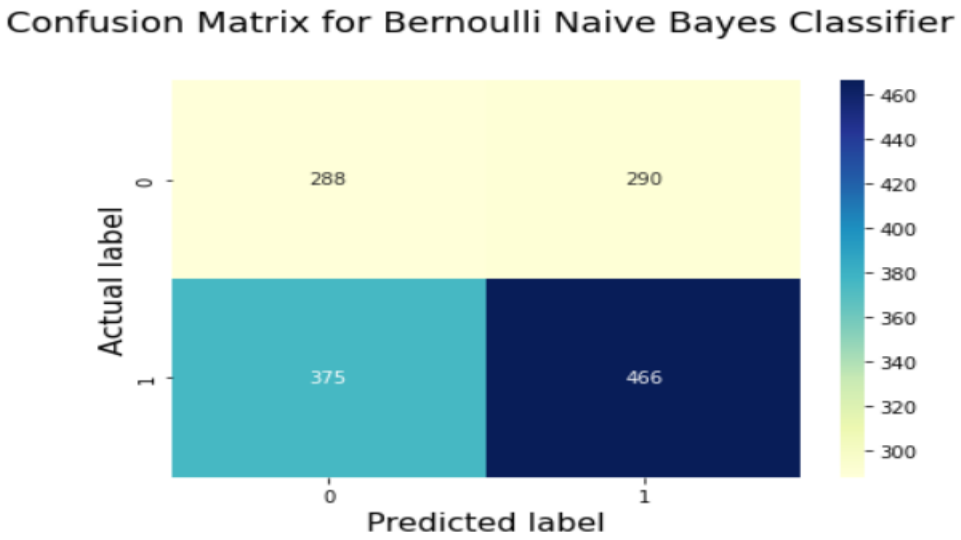

Fig 25: Confusion Matrix for Bernoulli Naïve Bayes Classifier

Classification Report for Bernoulli Naive Bayes Classifier is :

	precision	recall	f1-score	support
0.0	0.43	0.50	0.46	578
1.0	0.62	0.55	0.58	841
accuracy			0.53	1419
macro avg	0.53	0.53	0.52	1419
weighted avg	0.54	0.53	0.53	1419

Fig 26: Classification Report for Bernoulli Naïve Bayes Classifier

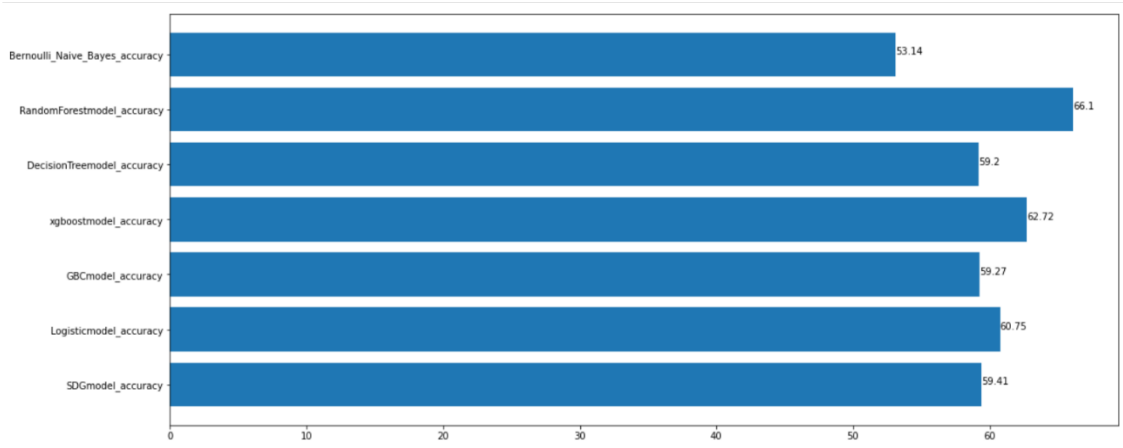

Fig 27: Accuracy Comparisons for the Classifiers

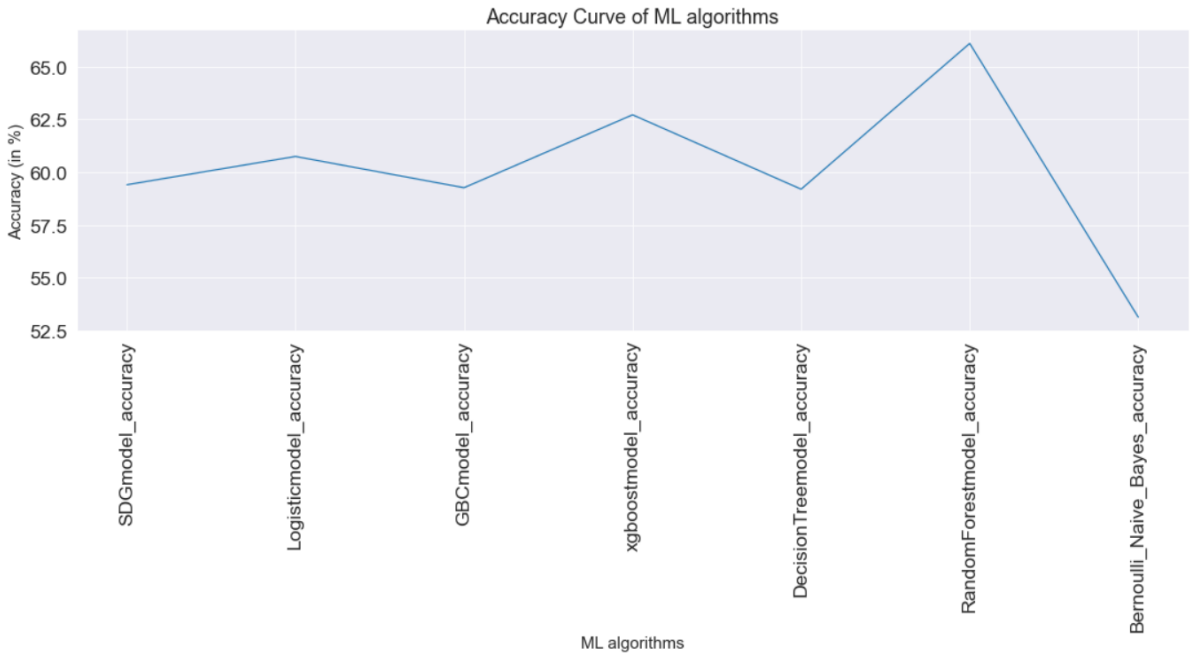

Fig 28: Accuracy Curve of ML Algorithms

The above graph shows that the data pipeline doesn't solve our concerns related to accuracy, so we will now move on to our Deep Learning Models in order to build a better classifier which will justify our desired results.

5.4.2 Deep Learning Models

a. SVM:

SVMs [17] are a type of algorithm that can be used for a variety of purposes, including classification, regression, and others. These algorithms produce either a single hyper-plane or a collection of hyper-planes in a space that has a high or an infinitely high number of dimensions. The distance that is the farthest from the nearest training data point for each class is the functional margin of the hyperplane that intuitively achieves a good separation. This distance is what distinguishes one class from another. This is due to the fact that the generalisation error of the classifier will decrease as the margin gets larger.

The Support Vector Machine, also known as SVM, was developed by fusing together two separate algorithms, namely:

i. SVC (When the hyperplane performs its function linearly, the classification method is referred to as SVC.)
ii. Linear SVC (The algorithm that separates the dataset by known linear approach then we call it as SVM.)

When it comes to fine-tuning the hyper-parameters of the two algorithms that were just explained, we make use of a wide variety of cross-validation values (CV). We were successful in accomplishing our objectives as a result of improvements in precision and recall, in addition to a general reduction in error rates. The Kernel = RBF algorithm is utilised in order to perform the fine-tuning of the SVC hyperparameters.

The linear SVC can be represented by the following expression: We employ two distinct methodologies in order to put the algorithm through its paces: first, a Linear SVC model, and then a standardised scale. These methodologies are used in this order. These methodological procedures are carried out in the order listed above. After that, we perform hyper-parameter tuning on this model by using Grid Search CV, Randomize Search CV, and a scaled feature. As a direct consequence of this, we were successful in accomplishing the required degree of precision while preserving an exceptionally low error rate.

The same algorithm is then utilised on a number of different cross-validation values, such as cv = 3, 6, and 8. In order for us to accomplish this goal, we can put our algorithm to the test by applying it to a number of different cross-validation values to see how well it does. Throughout the entire process, we were able to accomplish our goals while preserving a high degree of accuracy.

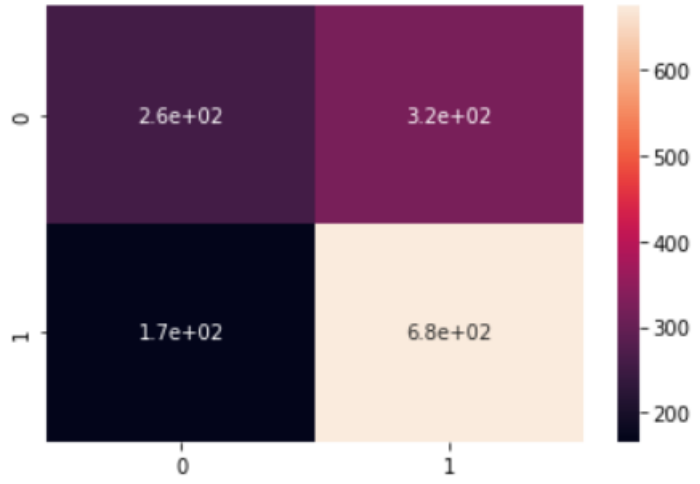

Fig 29: Confusion Matrix for Simple SVC Model

Classification report for train data is :

	precision	recall	f1-score	support
0.0	0.87	0.80	0.83	2279
1.0	0.87	0.92	0.90	3394
accuracy			0.87	5673
macro avg	0.87	0.86	0.86	5673
weighted avg	0.87	0.87	0.87	5673

Classification report for test data is :

	precision	recall	f1-score	support
0.0	0.61	0.44	0.51	578
1.0	0.68	0.80	0.73	841
accuracy			0.66	1419
macro avg	0.64	0.62	0.62	1419
weighted avg	0.65	0.66	0.64	1419

Fig 30: Classification Report of Simple SVC Model

```
from sklearn.model_selection import GridSearchCV
from sklearn.svm import SVC
model_parameters = {'C': [0.1, 1, 10, 100, 1000],
              'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
              'kernel': ['rbf']}
```

```
gscv = GridSearchCV(SVC(),
                    param_grid = model_parameters,
                    cv = 7,
                    verbose = 3,
                    n_jobs = -1)

gscv.fit(X_train_vall, Y_train_vall)
```

```
Fitting 7 folds for each of 25 candidates, totalling 175 fits
GridSearchCV(cv=7, estimator=SVC(), n_jobs=-1,
             param_grid={'C': [0.1, 1, 10, 100, 1000],
                         'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
                         'kernel': ['rbf']},
             verbose=3)
```

Fig 31: Hyper-parameter tuning of Simple SVC Model

```
Classification report for train data is :
              precision    recall  f1-score   support

         0.0       0.00      0.00      0.00        86
         1.0       0.62      1.00      0.77       141

    accuracy                           0.62       227
   macro avg       0.31      0.50      0.38       227
weighted avg       0.39      0.62      0.48       227


Classification report for test data is :
              precision    recall  f1-score   support

         0.0       0.00      0.00      0.00        26
         1.0       0.54      1.00      0.70        31

    accuracy                           0.54        57
   macro avg       0.27      0.50      0.35        57
weighted avg       0.30      0.54      0.38        57
```

Fig 32: Classification Report of Hyper-parameter tuned Simple SVC Model

Classification report for test data is :

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	2279
1.0	1.00	1.00	1.00	3394
accuracy			1.00	5673
macro avg	1.00	1.00	1.00	5673
weighted avg	1.00	1.00	1.00	5673

Fig 33: Classification report of Linear SVC Model

Classification report for test data is :

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	2279
1.0	1.00	1.00	1.00	3394
accuracy			1.00	5673
macro avg	1.00	1.00	1.00	5673
weighted avg	1.00	1.00	1.00	5673

Fig 34: Classification report of Standardized Linear SVC Model

b. CNN (Convolutional Neural Network):

Convolutional, pooling, and fully connected (FC) layers constitute the architecture of a CNN model [50, 51]. Convolutional layers of 2D-CNN models contain a total of 32 3x3-dimensional filters. In addition, activation functions such as ReLU, Sigmoid, and Tanh have been added to convolutional layers. In the "Experiment Results" section of the report, the outcomes of these various activation functions will be compared to one another. To classify the data, dense layers were utilised to convert matrices to vectors, flatten layers were utilised to convert matrices to vectors, max-pooling layers were utilised to reduce the dimensions of the data, dropout layers with variable dropout rates were utilised to prevent overfitting, and dropout layers with variable dropout rates were utilised to prevent overfitting. Utilizing the sigmoid activation function, which is present in the dense layer that follows the dense one, it is possible to classify data into two categories. Following that, the 317 dense layers will be utilised.

The number of filters, the size of the kernel, and the activation function determine the unique properties of the second proposed model for a 2D-CNN with three convolutional layers. The convolutional portion of this model's representation contains 5000 neurons. In addition, the model is comprised of one flat layer and two dense layers. The initial dense layer's activation

function is of the ReLU type, whereas the final dense layer's activation function is for sigmoid classification.

The initial step is to normalise our data, followed by separating it into test and training sets, and the final step is to run the 2D-CNN with as little cooling and dropouts as possible. Tanh, Sigmoid, and ReLU activation functions are utilised during the initial phase of the process when working with a binary cross entropy loss function.

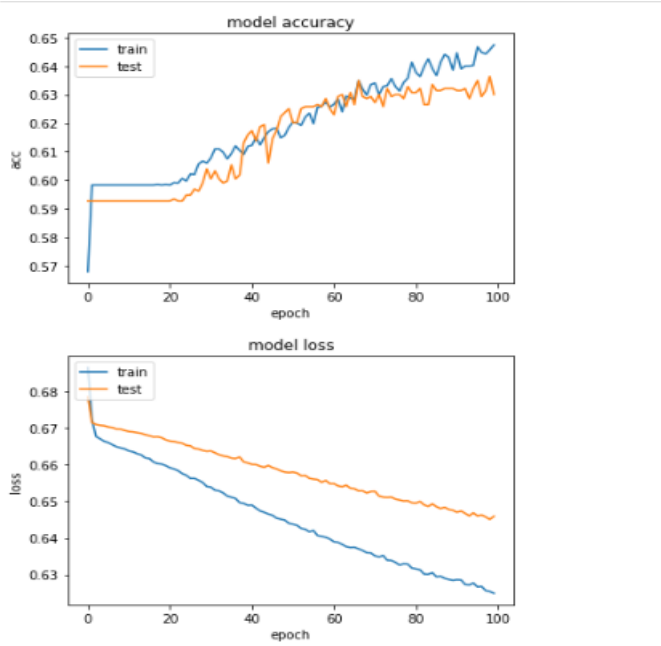

Fig 35: Model Accuracy and Model Loss of CNN

```
y_predicted =  model.predict(X_test_2d)

confusion_matr = confusion_matrix(Y_test_norm, np.round(model.predict(X_test_2d)))
confusion_matr
```

```
array([[150, 428],
       [ 97, 744]], dtype=int64)
```

```
print(
 "True Positive for schizophrenia ", confusion_matr[0, 0], "\n",
 "False Positive for schizophrenia ", confusion_matr[0, 1], "\n",
 "False Neagtive For Healthy ", confusion_matr[1, 0], "\n",
 "True Neagtive For Healthy ", confusion_matr[1, 1], "\n")
```

```
True Positive for schizophrenia  150
 False Positive for schizophrenia  428
 False Neagtive For Healthy  97
 True Neagtive For Healthy  744
```

```
TP = confusion_matr[0, 0]
FP = confusion_matr[0, 1]
FN = confusion_matr[1, 0]
TN = confusion_matr[1, 1]
TP, FP, FN, TN
```

```
(150, 428, 97, 744)
```

```
sensitivity = TP / (TP + FN)
sensitivity
```

```
0.607287449392712S
```

```
specificity = TN / (TN + FP)
specificity
```

```
0.6348122866894198
```

Fig 36: Classification Report of CNN

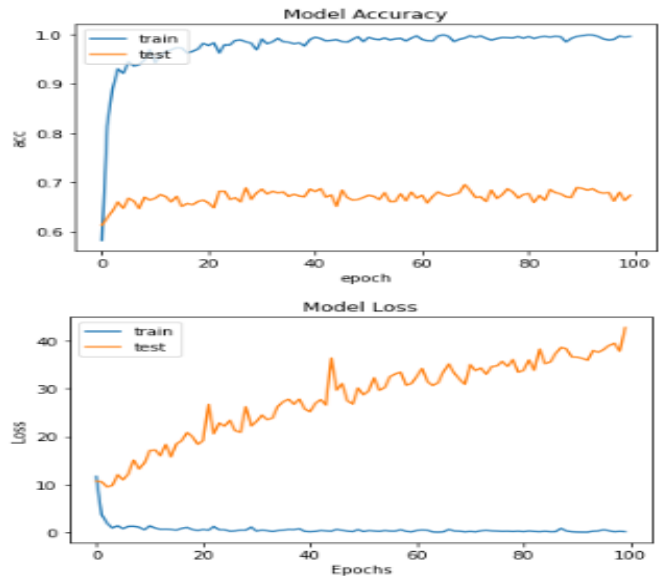

Fig 37: Second CNN Model Accuracy and Model Loss

```
y_predicted =  model_2.predict(X_test)

confusion_matr = confusion_matrix(Y_test, np.round(model_2.predict(X_test)))
confusion_matr
```

```
array([[289, 289],
       [174, 667]], dtype=int64)
```

```
print(
 "True Positive for schizophrenia ", confusion_matr[0, 0], "\n",
 "False Positive for schizophrenia ", confusion_matr[0, 1], "\n",
 "False Neagtive For Healthy ", confusion_matr[1, 0], "\n",
 "True Neagtive For Healthy ", confusion_matr[1, 1], "\n"
)
```

```
True Positive for schizophrenia  289
 False Positive for schizophrenia  289
 False Neagtive For Healthy  174
 True Neagtive For Healthy  667
```

```
TP = confusion_matr[0, 0]
FP = confusion_matr[0, 1]
FN = confusion_matr[1, 0]
TN = confusion_matr[1, 1]
TP, FP, FN, TN
```

```
(289, 289, 174, 667)
```

```
sensitivity = TP / (TP + FN)
sensitivity
```

```
0.6241900647948164
```

```
specificity = TN / (TN + FP)
specificity
```

```
0.6976987447698745
```

Fig 38: Classification Report for Second CNN Model

c. LSTM models

Speech recognition [53], natural language processing (NLP) [54], and biomedical signal processing [55, 56] are some of the applications that make use of recurrent neural networks. (RNNs). The designs used by CNN are completely Feed-Forward. In contrast, RNNs have a FeedBack layer that returns the network output in addition to the subsequent input. This is in contrast to the fact that there is no such layer in CNNs. RNNs are able to recall previously received information and apply it to the processing of subsequent inputs as a result of their internal memory. This ability allows RNNs to improve their accuracy. According to [52], the three primary RNN subtypes are referred to as simple RNN networks, LSTM networks, and GRU networks. In this section, several LSTM models for determining if someone has schizophrenia based on their EEG signals are proposed.

i. The first version of LSTM model

The first proposed LSTM model, is made up of three different layers Front(Input) gates, Forget Gate and Output Gate. An LSTM layer consisting of 128-neurons is utilised in the construction of this model. These layers are as follows: 3 dropout layers, 10 dense layers. The ReLU and sigmoid activation functions are put to use.

iii. The second version of LSTM model

The second LSTM model, it is composed of three layers again similar to the first version. In this architecture, LSTM layer consists of 100-neurons is utilised. When EEG signals are used as the data source for this study, the purpose of this investigation is to determine how the addition of LSTM layers to SZ affects the diagnostic accuracy of the model.

The LSTM consists of three layers, each of which contains a dropout. The total number of neurons is 100 for our second model, we also use an extra embedding layer apart from the three layers. In this LSTM model our dense layer is 1.

The dataset is then divided in 80:20 ration, with 80% becoming the Training Data and the other 20% becoming the Testing Data. At some point during the training procedure, the three-layer LSTM model is applied to the Training Data. Before training and testing can be performed, it is necessary to reformat the input data dimension. This can be achieved by setting the dimension to 64. The binary cross entropy is used as the basis for the loss function, which is realised by employing the RMSProp optimizer during the process of adjusting the learning rate. In the updated version of the model, a dense output layer based on softmax has been included.

The binary cross-entropy is then employed as the loss function and the model is compiled using the Adam optimizer for a total of two hundred iterations. Using the Testing Data, the precision, recall, f1-score, and loss of the model are evaluated in the final step. At this stage, the effectiveness of the model will be evaluated.

When evaluating a system, training and validation can be time-consuming processes; in fact, the size of the dataset after conversion (to integer format) may be smaller than the original text or string contained in the packets. This is due to the fact that the space required by the integer data type may be less than that required by the text data type. On the other hand, although unlikely, there is a remote possibility that the size will decrease. As a result, the number of words that will be trained on can have a substantial effect on the amount of time required to complete the training. For the purposes of this study, we have considered the full dataset, to get our model train better. We performed this algorithm on standard value but unfortunately didn't get the required output/ accuracy.

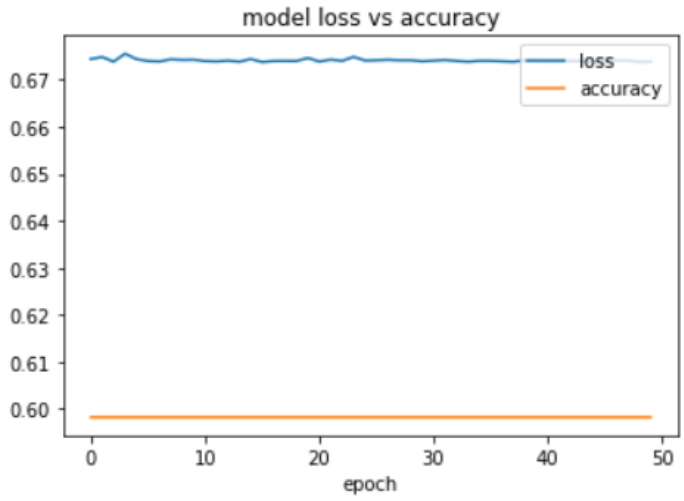

Fig 39: Model Loss vs Accuracy of LSTM Model

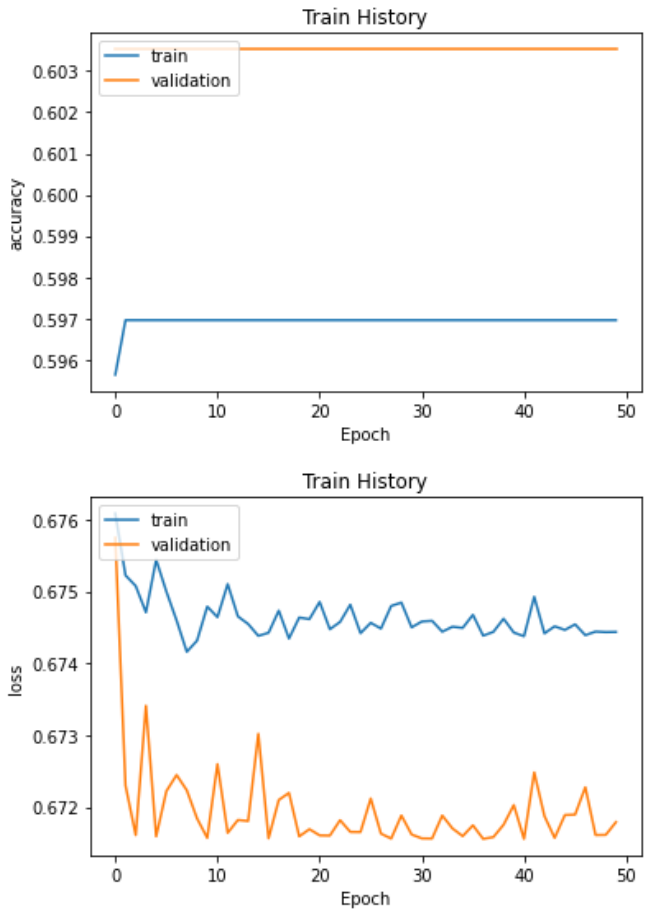

Fig 40: Train and Test Data Accuracy on different Epoch for Second Version of LSTM

d. ***CNN*-LSTM models**:

In CNN-RNN models, the first layer will typically consist of convolutional layers [52]. These layers are used to extract features and identify local patterns. The outputs from preceding RNN stages are utilised by subsequent layers. Experiments show that convolutional layers perform significantly better than RNNs when it comes to extracting both local and spatial EEG signal patterns. An RNN that makes use of convolutional layers is able to perform data analysis with greater precision. Several CNN-LSTM models for SZ diagnosis are going to be proposed in the section that comes after this one.

i. **The first version of CNN-LSTM model**

The CNN-LSTM model that has been proposed contains a total of five layers, some of which are referred to as embedding, dropout, CNN, LSTM, and dense. Convolutional layer with 32

filters, LSTM layer with 100 neurons, and one dense layer with sigmoid activation functions are included in this architecture.

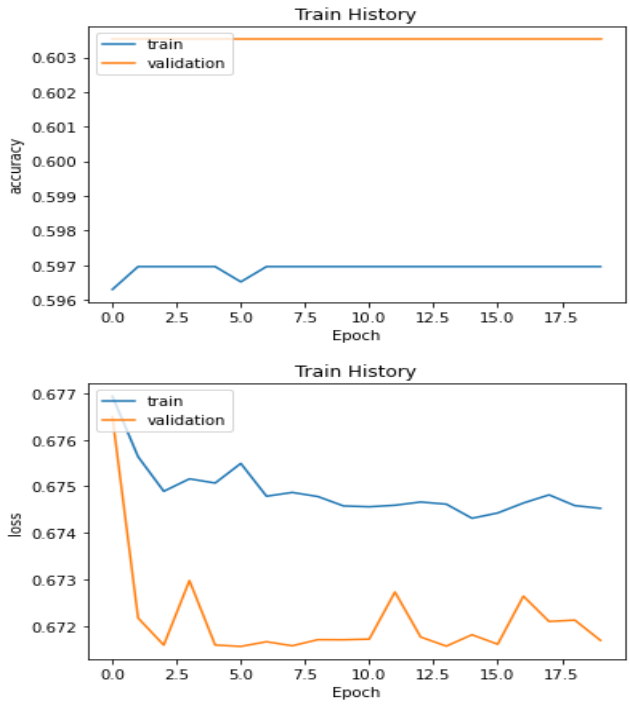

Fig 41: Train and Test Accuracy on different Epoch for First version of CNN-LSTM Model

ii. The second version of CNN-LSTM model

In the following sections, we will discuss the second CNN-LSTM model that has been proposed. The CNN and LSTM layers of this network are CNN, Embedding, LSTM, dense. In the dense layer of the architecture, the ReLU activation function and a dense layer of 100 neurons are utilised. This layer is an essential component of the building's structure. The dense layer is utilised for classification purposes.

The following steps are required prior to feeding our CNN-LSTM model tensors. The features are separated into train, validation, and test sets based on their respective requirements and the model is trained and tested first on CNN and then consequently on LSTM.

```
None
Epoch 1/5
71/71 - 23s - loss: 0.7546 - accuracy: 0.5696 - val_loss: 0.6764 - val_accuracy: 0.6035
Epoch 2/5
71/71 - 22s - loss: 0.6757 - accuracy: 0.5970 - val_loss: 0.6726 - val_accuracy: 0.6035
Epoch 3/5
71/71 - 22s - loss: 0.6771 - accuracy: 0.5970 - val_loss: 0.6716 - val_accuracy: 0.6035
Epoch 4/5
71/71 - 22s - loss: 0.6760 - accuracy: 0.5970 - val_loss: 0.6719 - val_accuracy: 0.6035
Epoch 5/5
71/71 - 22s - loss: 0.6777 - accuracy: 0.5970 - val_loss: 0.6726 - val_accuracy: 0.6035
```

Fig 42: Second Version of CNN-LSTM Model

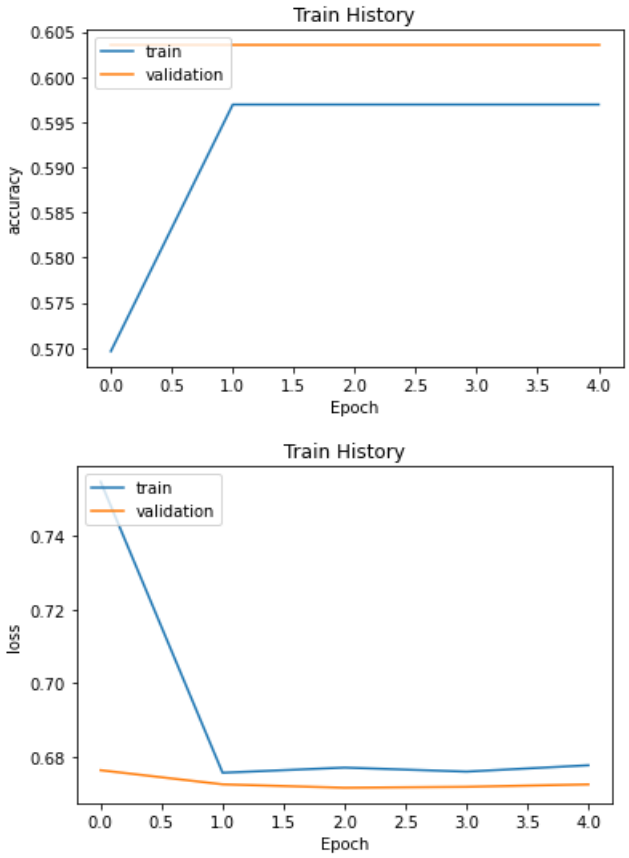

Fig 43: Train and Test Data accuracy on different Epoch for Second Version of CNN-LSTM Model

Table 2: Accuracy Score for Deep Learning Classification Models

Model	Kernels	Parameters Settings	Accuracy	Error
SVM-SVC	RBF	C = 1.0, 'gamma' = Scale	87.18/65.60	MSE = 0.34
SVM-SVC (Hyper Parameter Tuning)	RBF	'C': [0.1, 1, 10, 100, 1000], 'gamma':[1, 0.1, 0.01, 0.001, 0.0001]	62.11/54.38	MSE = 0.94
kaLinear SVC(CV=3)	RBF	"C": uniform(1, 10), "gamma": reciprocal(0.001, 0.1)	100/99.98	MSE = 0
Standardized Linear SVC (CV=3)	RBF	"C": uniform(1, 10), "gamma": reciprocal(0.001, 0.1)	100/100	MSE = 0
Linear SVC (CV=6)	RBF	"C": uniform(1, 10), "gamma": reciprocal(0.001, 0.1)	100/99.98	MSE = 0
Standardized Linear SVC (CV=6)	RBF	"C": uniform(1, 10), "gamma": reciprocal(0.001, 0.1)	100/100	MSE = 0
Linear SVC (CV=8)	RBF	"C": uniform(1, 10), "gamma": reciprocal(0.001, 0.1)	100/99.98	MSE = 0

Standardized Linear SVC (CV=8)	RBF	"C": uniform(1, 10), "gamma": reciprocal(0.001, 0.1)	100/100	MSE = 0
CNN (ReLU)	Conv2d, MaxPooling2D	kernel_size=(5, 20)/ (3,3) activation='tanh','relu',' sigmoid'	64.75/63.63	Loss = 0.64
CNN		'neurons'= 5000, 'activation' = 'relu', 'sigmoid'	99.59/ 69.5	Loss = 0.42
LSTM (First Version)		'neurons' = 128 'activation' = 'sigmoid' 'dense' = 10	59.83/ 0.0000e+00	Loss = 0.67
LSTM (Second Version)		'neurons' = 100 'activation' = 'sigmoid' 'dense' = 1	59.70/ 60.35	Loss = 0.67
CNN-LSTM (First Version)		'neurons' = 100 'activation' = 'sigmoid' 'dense' = 1	59.69/ 60.35	Loss = 0.67
CNN-LSTM (Second Version)		'neurons' = 100 'activation' = 'relu' 'dense' = 1	59.69/60.35	Loss = 0.67

Table 3: Performance of SVM Classifier

Name	Accuracy	Precision	Recall	F1-Score
Standardized Linear SVC (CV=3)	100%	100%	100%	100%
Standardized Linear SVC (CV=6)	100%	100%	100%	100%
Standardized Linear SVC (CV=8)	100%	100%	100%	100%

CHAPTER 6

RESULTS AND DISCUSSION

SZ is a mental disorder that interferes with normal brain function and can result in a wide range of day-to-day difficulties for patients. For the purpose of diagnosing schizophrenia spectrum disorders, also known as SZ mental disorders, a number of screening methods have been developed, with the EEG functional imaging modality attracting the attention of neurologists and specialist physicians. A diagnosis of coma based on EEG signals has historically been notoriously difficult to attain. A number of research projects utilising the application of AI methods to the diagnosis of SZ and the interpretation of EEG signals have been successfully completed over the past few years. Using EEG signals, it is hoped that neurologists and other medical professionals will be able to diagnose SZ disorder more quickly and accurately. This article investigates the use of EEG signals in conjunction with a variety of AI-based diagnostic methods for schizophrenia. DL models [19], [20], [21], and [22] are combined with a variety of conventional ML techniques [22]. AI models that diagnose SZ by analysing EEG signals incorporate all of the steps of selecting a dataset, preprocessing that dataset, extracting and selecting features, and classifying the data. The purpose of this is to diagnose SZ. This study collected EEG data from a total of 14 participants, including both healthy participants and those with a diagnosis of schizophrenia [23]. This dataset contains EEG signals sampled at 250 Hz and distributed across 10 channels [23]. This dataset contains forty distinct channels in total. During the initial stage of the preprocessing phase, the EEG signals were divided into 25-second frames. The EEG signals were subsequently transformed using the z-score and z-score-L2 functions. During this portion of the recording, each EEG frame had a resolution of 196250 pixels. It is important to note that z-score and z-score-L2 normalisation methods were used to preprocess EEG signals for DL models. The classification algorithms we used were designed using the system with the features:

Processor - AMD Ryzen 5 5600X , Ram - 16GB, Hard Disk - 1 TB, SSD - 500GB, GPU - 8GB RTX 3060TI.

Multiple classification algorithms based on conventional machine learning were used to diagnose SZ by analysing EEG signals. This section investigates the use of normalised EEG signals as features within classification algorithms. This study employed the following classification algorithms: SVM [24], KNN [25], DT [26], naive Bayes [27], RF [28], ERT [29], and bagging [30]. In comparison to other classification methods, Simple SVC and Linear SVM

classification using EEG signals had the highest accuracy, with test and train accuracy of 100/100. The algorithms were evaluated using a technique known as cross validation with 3,6 and 8 folds. In the sections that follow, we will examine a variety of DL methods for diagnosing schizophrenia based on EEG signals. This section describes a variety of deep learning strategies, including the ones listed below: There are two LSTM models, two 2D-CNN structures, and two 2D-CNN-LSTM networks. Various activation functions, such as ReLU, Sigmoid, and tanh, were used to implement the proposed deep learning models. Throughout the classification process, each model utilised its own implementation of the sigmoid activation function. The results of a re-evaluation of the outcomes predicted by DL models employing a variety of distinct normalisation strategies and activation functions are presented in Tables below. The fact that this model has never achieved an accuracy of 100 percent with training of full dataset compared to previous paper studies is unique. Figure below compares our best Classification model to other models and used to diagnose SZ using EEG signals in our work.

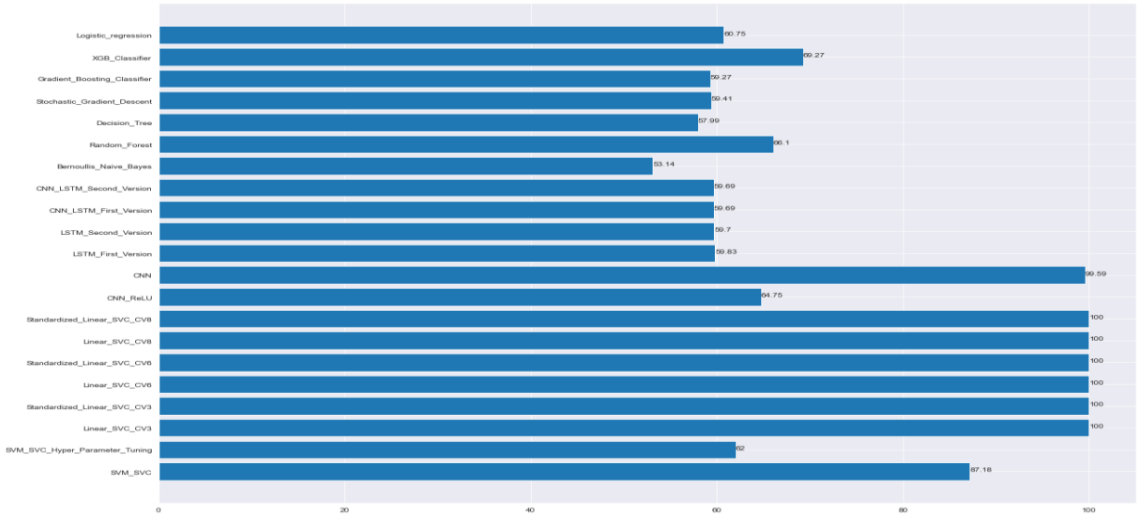

Fig 44: Accuracy Comparison for Machine Learning and Deep Learning Models

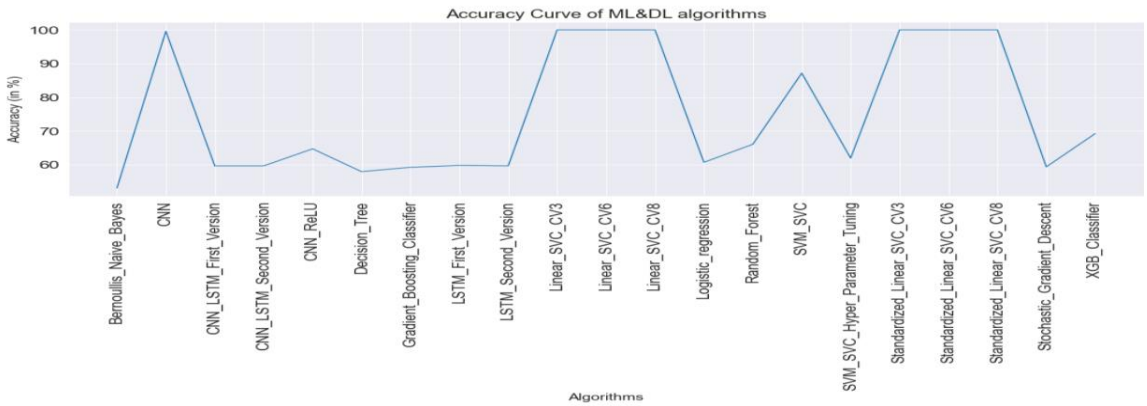

Fig 45: Accuracy Curve of Machine Learning and Deep learning Algorithms

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 Conclusion:

This dissertation compares the system performance of Decision Tree, Random Forest, SVM, Nave Bayes, XGBoost, LSTM, CNN, and LSTM-CNN prediction models developed for hospitalised schizophrenia patients. The study determined that SVM (accuracy = 100%) is the most accurate model for classifying hospitalised patients with schizophrenia into this group, followed by CNN (accuracy = 99.59%). We also tried training our model on other algorithms such as LSTM, CNN-LSTM, Random Forest and Decision Tree but due to overfitting and bad accuracy couldn't continue with them. We even tried data pipeline methods to avoid leakage of data in order to improve the accuracy of our test data, but it was not an effective approach. XGBoost did give us somewhat better number(accuracy=69%) when compared with other machine learning models but the run time for the continued for more than 2 days and eventually we had to stop the trial. Apart from all the approaches, the SVM and CNN algorithms perform better in terms of performance metrics than the other algorithm. The results indicate that algorithms such as SVM may assist hospitals in treating this disorder. In this regard, the predictive modelling approach could prevent the hospitalisation of individuals with schizophrenia. Consequently, we will concentrate on utilising future lines to predict risk factors in hospitalised patients with schizophrenia and readmissions in the acute units of each health complex.

complex.

Table 4: A comparison of related work with our model accuracy and best classifier.

Work Reference	Techniques Used	Best Performing Classifier	Overall Accuracy	Dataset	EEG Data (Test Subjects)
[36]	SVM	SVM with 60 features and cross validation to validate.	87.0%	Public Dataset	40 Schizophrenia and 12 healthy subjects
[31]	DT, LD, KNN, PNN, SVM	SVM using RBF kernel and t-test for feature selection	92.90	Institute of Psychiatry and	14 Schizophrenia and 14

				Neurology in Warsaw, Poland	healthy subjects
[28]	SVM	SVM with combined feature set of 15	88.24%	Clinical Dataset	34 schizophrenia patients and 34 healthy subjects
[39]	SVM, Multilayer perceptron (MLP) classifiers	MLP using feature selection J5	93.42%	Public Dataset	31 healthy subjects and 16 patients
[8]	CNN	CNN with 11 layes and ReLU activation function	89.59	Institute of Psychiatry and Neurology in Warsaw, Poland	14 healthy subject and 14 schizophrenia patients
[40]	Feed Forward Neural Network, SVM	SVM for EEG, Resting +two different mental task(visual stimulus) using RBF Kernel	88.50%	Clinical Trial	55 schizophrenia subjects and 23 healthy subjects
[10]	SVM , KNN, DT , Naïve Bayes, RF, 1D-CNN, 1D-CNN-LSTM	1D CNN-LSTM with 5-fold-cross validation using 11 Max, dropout, CNN, LSTM, flatten, pooling, and dense layers.	99.25%	Institute of Psychiatry and Neurology in Warsaw, Poland	Schizophrenia patients:14, Healthy patients:14
[9]	DT, KNN, EBT, SVM	EBT using Approximate Entropies empirical mode decomposition (EMD) based characteristics	89.59%	Kaggle	49 patients with schizophrenia and 32 healthy control subjects
[41]	SzNet-5(with Ensemble method), SzNet-15, SzNet-35(with Deep learning	SzNet-5(5 meaning 5 midline electrodes) with Ensemble method using cross-validation	78%	Kaggle	63 healthy patients and 65 schizophrenia patients

	method like CNN)	10 and hyper parameter tuning.			
[42]	Random Forest	Random Forest using 10-cross validation	96.77%	Public Dataset	14 healthy patients and 14 schizophrenia patients
[43]	Random Forest	Random Forest using 10-cross validation and mtry(number of features) = 8	81.10%	Kaggle	49 Schizophrenia patients and 32 healthy patients
[44]	TCN, MLPNN, TCN-AE	MLPNN using 7-fold cross validation	64%	Clinical Dataset	NA
[45]	LSTM, SNN	SNN method with 1-fold-cross validation	100%	Kaggle	32 Healthy subjects; 49 Schizophrenia patients.
Presented Work	Logistic Regression, RF, DT, XGBoost, SVM, 2D-CNN, LSTM,2D-CNN-LSTM	Standardized Linear SVM using 3,6,8 cross-validation with RBF Kernel	100%	Kaggle	32 Healthy subjects; 49 Schizophrenia patients.

7.2 Future Scope:

We plan to expand the scope of the experimental study to include additional disorders and populations, each of which may use a different instrumentations as well as protocols for EEG; compare the performance of the proposed methodologies for EEG data classification by comparing all of the models that have been mentioned; increase the accuracy while minimising the loss and error caused by using a variety of kernels, activation functions, and fitting dense layers; and develop a method that is relevant to a range of disorders as well as the populations. Future research will concentrate on combining traditional ML models with models based on the deep learning in order to diagnose schizophrenia by extracting various nonlinear EEG signal characteristics beforehand. Following this, DL models are used to extract features from the previously extracted raw EEG signals. The classification process is complete once the DL and handcrafted characteristics are combined. Utilizing graph models powered by deep learning is one of the novel approaches being investigated for the diagnosis of brain disorders (DL).

References:

[1] Wenya Nan, Joao Pedro Rodrigues, Jiali Ma, Xiaoting Qu, Feng Wan, Pui-In Mak, Peng Un Mak, ˜ Mang I Vai, and Agostinho Rosa. Individual alpha neurofeedback training effect on short term memory. International journal of psychophysiology, 86(1):83–87, 2012.

[2] Adrian J Fowle and Colin D Binnie. Uses and abuses of the eeg in epilepsy. Epilepsia, 41: S10–S18, 2000.

[3] Michael J Owen, Akira Sawa, and Preben B Mortensen. Schizophrenia. The Lancet, 388(10039):86 – 97, 2016.

[4] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sanchez. "A survey on deep learning in medical image analysis." Medical image analysis, 42:60–88, 2017.

[5] Fernando Lopes da Silva. Eeg and meg: relevance to neuroscience. Neuron, 80(5):1112–1128, 2013.

[6] A. Shoeibi, N. Ghassemi, R. Alizadehsani, M. Rouhani, H. HosseiniNejad, A. Khosravi, M. Panahiazar, and S. Nahavandi, "A comprehensive comparison of handcrafted features and convolutional autoencoders for epileptic seizures detection in eeg signals," Expert Systems with Applications, vol. 163, p. 113788, 2021.

[7] A. Shalbaf, S. Bagherzadeh, and A. Maghsoudi, "Transfer learning with deep convolutional neural network for automated detection of schizophrenia from eeg signals," Physical and Engineering Sciences in Medicine, vol. 43, no. 4, pp. 1229–1239, 2020.

[8] S. L. Oh, J. Vicnesh, E. J. Ciaccio, R. Yuvaraj, and U. R. Acharya, "Deep convolutional neural network model for automated diagnosis of schizophrenia using eeg signals," Applied Sciences, vol. 9, no. 14, p. 2870, 2019.

[9] Siuly Siuly, Smith K. Khare, Varun Bajaj, Hua Wang and Yanchun Zhang, "A Computerized Method for Automatic Detection of Schizophrenia Using EEG Signals", Transactions on Neural Systems and Rehabilitation Engineering, 2020.

[10] Afshin Shoeibi, Delaram Sadeghi, Parisa Moridian, Navid Ghassemi, Jonathan Heras, Roohallah Alizadehsani, Ali Khadem, Yinan Kong, Saeid Nahavandi, Yu-Dong Zhang, and

Juan M. Gorriz, “Automatic Diagnosis of Schizophrenia in EEG Signals Using CNN-LSTM Models”, arXiv:2109.01120, 2021.

[11] T. Naira and C. Alberto, “Classification of people who suffer schizophrenia and healthy people by eeg signals using deep learning,” 2020.

[12] D. Ahmedt-Aristizabal, T. Fernando, S. Denman, J. E. Robinson, S. Sridharan, P. J. Johnston, K. R. Laurens, and C. Fookes, “Identification of children at risk of schizophrenia via deep learning and eeg responses,” IEEE journal of biomedical and health informatics, vol. 25, no. 1, pp. 69–76, 2020.

[13] J. Sun, R. Cao, M. Zhou, W. Hussain, B. Wang, J. Xue, and J. Xiang, “A hybrid deep neural network for classification of schizophrenia using eeg data,” Scientific Reports, vol. 11, no. 1, pp. 1–16, 2021.

[14] G. Sharma, A. Parashar, and A. M. Joshi, “Dephnn: A novel hybrid neural network for electroencephalogram (eeg)-based screening of depression,” Biomedical Signal Processing and Control, vol. 66, p. 102393, 2021.

[15] K. Singh, S. Singh, and J. Malhotra, “Spectral features based convolutional neural network for accurate and prompt identification of schizophrenic patients,” Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine, vol. 235, no. 2, pp. 167–184, 2021.

[16] Judith M. Ford, Vanessa A. Palzes, Brian J. Roach, and Daniel H. Mathalon , “Did I Do That? Abnormal Predictive Processes in Schizophrenia When Button Pressing to Deliver a Tone”, Schizophr Bull. 2014 Jul; 40(4): 804–812.

[17] Durgesh K. Srivastava, Lekha Bhambhu, Data Classification Using Support Vector machine, Journal of Theoretical and Applied Information Technology, 2005-2009.

[18] J. M. Gorriz, J. Ram ´ ´ırez, A. Ort´ız, F. J. Martinez-Murcia, F. Segovia, J. Suckling, M. Leming, Y.-D. Zhang, J. R. Alvarez-S ´ anchez, ´ G. Bologna, et al., “Artificial intelligence within the interplay between natural and artificial computation: Advances in data science, trends and applications,” Neurocomputing, vol. 410, pp. 237–270, 2020.

[19] C. Jimenez-Mesa, J. Ram ´ ´ırez, J. Suckling, J. Voglein, J. Levin, J. M. ¨ Gorriz, A. D. N. I. ADNI, and D. I. A. N. DIAN, “Deep learning ´ in current neuroimaging: a multivariate

approach with power and type i error control but arguable generalization ability," arXiv preprint arXiv:2103.16685, 2021.

[20] J.-M. Gorriz, J. Suckling, J. Ramirez, C. Jimenez-Mesa, and F. Segovia, "A connection between pattern classification by machine learning and statistical inference with the general linear model," IEEE Journal of Biomedical and Health Informatics, 2021.

[21] F. J. Martinez-Murcia, A. Ortiz, J.-M. Gorriz, J. Ramirez, and D. Castillo-Barnes, "Studying the manifold structure of alzheimer's disease: A deep learning approach using convolutional autoencoders," IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 1, pp. 17–26, 2019.

[22] R. Alizadehsani, M. Roshanzamir, S. Hussain, A. Khosravi, A. Koohestani, M. H. Zangooei, M. Abdar, A. Beykikhoshk, A. Shoeibi, A. Zare, et al., "Handling of uncertainty in medical data using machine learning and probability theory techniques: A review of 30 years (1991–2020)," Annals of Operations Research, pp. 1–42, 2021.

[23] E. Olejarczyk and W. Jernajczyk, "Graph-based analysis of brain connectivity in schizophrenia," PLoS One, vol. 12, no. 11, p. e0188629, 2017.

[24] C. Cortes and V. Vapnik, "Support-vector networks," Machine learning, vol. 20, no. 3, pp. 273–297, 1995.

[25] T. Cover and P. Hart, "Nearest neighbor pattern classification," IEEE transactions on information theory, vol. 13, no. 1, pp. 21–27, 1967.

[26] L. Rokach and O. Z. Maimon, Data mining with decision trees: theory and applications. World scientific, 2007, vol. 69.

[27] H. Zhang, "The optimality of naive bayes," AA, vol. 1, no. 2, p. 3, 2004. [45] L. Breiman, "Random forests," Machine learning, vol. 45, no. 1, pp. 5–32, 2001.

[28] M. Shim, H-J Hwang, Do-Won Kim, S-H Lee , C-H Im, "Machinelearning-based diagnosis of schizophrenia using combined sensor-level and source-level EEG features." Schizophrenia research, vol. 176, no. 2-3, pp.314- 319, 2016.

[29] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," Annals of statistics, pp. 1189–1232, 2001.

[30] Amin Khorrama , Mohammad Khalooeib , Mansoor Rezghic, “End-to-End CNN+LSTM Deep Learning Approach for Bearing Fault Diagnosis”, *Appl Intell* **51,** 736–751 (2021).

[31] V Jahmunah, Shu Lih Oh, V Rajinikanth, Edward J Ciaccio, Kang Hao Cheong, N Arunkumar, U Rajendra Acharya, “ Automated detection of schizophrenia using nonlinear signal processing methods”, Artificial Intelligence In Medicine (2019).

[32] Winda Kurnia Sari , Dian Palupi Rini , Reza Firsandaya Malik, “Text Classification Using Long Short-Term Memory with GloVe Features”, Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI) Vol. 5, No. 1, June 2019, pp. 85~100

[33] Ren-Hung Hwang, Min-Chun Peng, Van-Linh Nguyen and Yu-Lun Chang, “An LSTM-Based Deep Learning Approach for Classifying Malicious Traffic at the Packet Level”, Applied Sciences, 2019.

[34] Manuel A. Vázquez, Arash Maghsoudi, and Inés P. Mariño, “An Interpretable Machine Learning Method for the Detection of Schizophrenia Using EEG Signals”, Front Syst Neurosci. 2021; 15: 652662.

[35] Susel Góngora Alonso, Gonçalo Marques, Deevyankar Agarwal, Isabel De la Torre Díez and Manuel Franco-Martín, “Comparison of Machine Learning Algorithms in the Prediction of Hospitalized Patients with Schizophrenia”, Sensors 2022, 22, 2517.

[36] Jason K. Johannesen, Jinbo Bi , Ruhua Jiang , Joshua G. Kenney and Chi-Ming A. Chen, “Machine learning identification of EEG features predicting working memory performance in schizophrenia and healthy adults”, Johannesen et al. Neuropsychiatric Electrophysiology (2016) 2:3.

[37] Farhana Sultana, Abu Sufian, Paramartha Dutta, “Advancements in Image Classification using Convolutional Neural Network”, 2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks.

[38] P. Geurts, D. Ernst, and L. Wehenkel, “Extremely randomized trees,” Machine learning, vol. 63, no. 1, pp. 3–42, 2006.

[39] L. Santos-Mayo, L. M. San-José-Revuelta, and J. I. Arribas, “A Computer-Aided Diagnosis System With EEG Based on the P3b Wave During an Auditory Odd-Ball Task in Schizophrenia,” IEEE Trans. Biomed. Eng., vol. 64, no. 2, pp. 395–407, 2017.

[40] B. Thilakvathi, S. D. Shenbaga, K. Bhanu, M. Malaippan, “EEG signal complexity analysis for schizophrenia during rest and mental activity.” Biomedical Research, vol. 28 (1), pp.1-9, 2017