DEEP LEARNING FOR THE CLASSIFICATION OF BIPOLAR DISORDER USING FNIRS MEASUREMENTS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HALUK BARKIN EVGIN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

FEBRUARY 2021

Approval of the thesis:

DEEP LEARNING FOR THE CLASSIFICATION OF BIPOLAR DISORDER USING FNIRS MEASUREMENTS

submitted by **HALUK BARKIN EVGIN** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. İlkay Ulusoy
Head of Department, **Electrical and Electronics Engineering** _____

Prof. Dr. İlkay Ulusoy
Supervisor, **Electrical and Electronics Engineering, METU** _____

Examining Committee Members:

Prof. Dr. Nevzat Güneri Gençer
Electrical and Electronics Engineering, METU _____

Prof. Dr. İlkay Ulusoy
Electrical and Electronics Engineering, METU _____

Prof. Dr. Uğur Halıcı
Electrical and Electronics Engineering, METU _____

Assoc. Prof. Dr. Yeşim Serinağaoğlu Doğrusöz
Electrical and Electronics Engineering, METU _____

Assoc. Prof. Dr. Bora Başkak
School of Medicine, Ankara University _____

Date: 03.02.2021

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Haluk Barkın Evgin

Signature :

ABSTRACT

DEEP LEARNING FOR THE CLASSIFICATION OF BIPOLAR DISORDER USING FNIRS MEASUREMENTS

Evgin, Haluk Barkın
M.S., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. İlkay Ulusoy

February 2021, 66 pages

Functional Near-Infrared Spectroscopy (fNIRS) is a neural imaging method that is proved to be prominent in the classification of psychiatric disorders, and assertive accuracy results are being obtained using fNIRS. High temporal resolution, feasibility, and partial endurance to head movements are the traits that are highlighting fNIRS among other imaging methods. fNIRS data is a one dimensional multi-channeled time series. In this thesis, bipolar disorder is classified using some state of the art deep learning methods that are specialized for time series classification. Multilayer Perceptrons, one dimensional Convolutional Neural Networks (CNN), one dimensional Residual Neural Networks (ResNet) and one dimensional Encoder networks are trained, evaluated and compared on the fNIRS data where there are 33 control and 28 bipolar subjects. Although the number of subjects is not high enough, promising accuracies are obtained using different test methods. The best classification accuracy of 75.32% is obtained by using the ResNet classifier.

Keywords: fNIRS, Deep Learning, Bipolar Disease, Classification

ÖZ

BİPOLAR BOZUKLUĞUN FNIRS ÖLÇÜMLERİ KULLANILARAK DERİN ÖĞRENME İLE SINIFLANDIRILMASI

Evgin, Haluk Barkın
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. İlkay Ulusoy

Şubat 2021 , 66 sayfa

İşlevsel Yakın İnfrared Spektroskopi (fNIRS) psikiyatrik bozuklukların sınıflandırılmasında öne çıkan bir sinirsel görüntüleme tekniğidir ve fNIRS kullanılarak psikiyatrik bozuklukların sınıflandırılmasında güvenilir doğruluk sonuçları elde edilmektedir. Yüksek zamansal çözünürlük, uygulanabilirlik ve kafa hareketlerine karşı kısmi direnç, fNIRS metodunu diğer görüntüleme metodları arasında öne çıkarmaktadır. fNIRS verisi tek boyutlu çok kanallı bir zaman serisidir. Bu tezde bipolar bozukluk, zaman serisi sınıflandırılmasında özelleşmiş en gelişkin metodlardan bazıları kullanılarak sınıflandırılmıştır. Çok Katmanlı Algılayıcılar (MLP), Tek Boyutlu Konvolüsyonel Sinir Ağları (CNN), Tek Boyutlu Rezidüel Sinir Ağları (ResNet) ve Kodlayıcı Ağlar, içerisinde 33 kontrol ve 28 bipolar örnek olan fNIRS verileri üzerinde eğitilmiş, değerlendirilmiş ve kıyaslanmıştır. Eldeki örnek sayısı yeterince fazla olmamasına rağmen farklı test metodları kullanılarak ümit verici doğruluk oranları elde edilmiştir. 75.32%'lik en iyi sınıflandırma doğruluğu, ResNet sınıflandırıcısı kullanılarak elde edilmiştir.

Anahtar Kelimeler: fNIRS, Derin Öğrenme, Bipolar Bozukluk, Sınıflandırma

To the memory of my father

ACKNOWLEDGMENTS

Firstly, I would like to express my deepest gratitude to my advisor Prof. Dr. Ilkay ULUSOY for her continuous and valuable guidance, support and encouragement. Her motivation and passion inspired me to proceed my research.

I would also like to thank Prof. Dr. Nevzat Güneri GENÇER, Prof. Dr. Uğur HALICI, Assoc. Prof. Dr. Yeşim SERİNAĞAOĞLU and Assoc. Prof. Dr. Bora BAŞKAK being in my jury and sharing their opinions. It has been a great honour and luck having these people I admire since my undergraduate studies.

I am so grateful to my family and especially to my father that I lost recently. Nothing would be possible without their endless support and love.

I would also like to thank my teammates Hülya KORKMAZ, Ufuk YILDIZ and Ali İbrahim ÖZKAN for their endless help; my study partner Oğuzhan BABACAN for the big support; and my girlfriend Gizem GÜVEN for always being by my side.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

fNIRS	Functional Near-Infrared Spectrography
ANN	Artificial Neural Networks
MLP	Multilayer Perceptron
CNN	Convolutional Neural Networks
fNIRS	Functional Near Infrared Spectrography
MLP	Multilayer Perceptron
ResNet	Residual Neural Network
GAP	Global Average Pooling
BD	Bipolar Disorder

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Definition

Using Deep Neural Networks to solve real-world problems is becoming more and more popular every day. Training neural networks on priorly obtained datasets helped computer scientists to solve problems by eliminating the need for expert knowledge of the data.

With the sharp increase in the technology of graphical processing units (GPUs) and tensor processing units (TPUs), deep learning and artificial intelligence-based methods are being used in the solutions for problems in almost every industrial area or even in real life problems instead of conventional machine learning approaches. Time series data classification is one of the fundamental areas that is being worked on extensively.

The aim of this thesis is to classify patients with Bipolar Disorder (BD) by using Functional Near-Infrared Spectroscopy (fNIRS) measurements with the help of some state of the art deep learning methods. fNIRS is a biomedical imaging method based on the emission and absorption of near-infrared lights on a subject's scalp using dedicated hardware consisting of emitters and detectors. As expected, computer scientists and researchers do not have adequate information to diagnose BD with the available data. However, the time series classification (TSC) using deep learning and artificial intelligence methods is a significant innovation recently researched. Apart from visual classification tasks, these networks are being specialized for TSC. Some of these networks are Multilayer Perceptrons, the most fundamental architecture of artificial intelligence; Convolutional Neural Networks (CNN), one of the most successful neu-

ral networks; Residual Neural Networks (ResNet), and Encoder networks, which are modified versions of CNN. Classifying the disease by applying the mentioned networks is a great time saving achievement for the doctors working on the classification of BD.

Measurements are taken from two groups of people. The first group consists of people who have BD, and the second group which is control group. Despite the inadequate expertise on specific biomedical imaging methods and the experimental data, it is expected to attain a tenable classification accuracy for diagnosing the disease by using the power of artificial and convolutional neural networks. As mentioned in chapter 2, the fNIRS data is a multi-channeled temporal sampled data; implementation of state of the art neural networks suggested in many kinds of research for multivariate time series classification is the primary goal of this thesis. The evaluation and the comparison in terms of the classification performance of the aforementioned neural networks for bipolar disease using fNIRS data is an under-researched area that should be enlightened in this thesis's scope.

1.2 Proposed Methods and Models

The starting point of this research is the use of MLP for the simple implementation of time series classification. MLP will form a basis for the research; however, MLP is not the network that is aimed to obtain the best results.

One-dimensional CNNs are the second neural network model that is being tried. Although CNNs are leading classifiers for visual classification tasks, they are often being used for TSC and are very successful [1].

To increase stacked layers on a deep neural network, some modifications such as ResNet [1] are suggested for TSC and the ResNets are the next models that are tried for classification of BD with fNIRS measurements.

After convolutional layers, neural networks require a layer to summarize the temporal structure obtained at the network's output. This layer is chosen for 1-dimensional CNNs and ResNets to be a Global Average Pooling layer (GAP). However, there are

some networks for TSC that replace the GAP layer with another architecture called attention layer. These networks are called as Encoder networks, and the last network suggested in the thesis is the Encoder networks.

1.3 Contributions and Novelties

In 2019, our research group has published the first paper and used one dimensional CNNs for classification of BD using fNIRS data [2]. This research was the only work that exercises the fNIRS response input for BD classification with deep learning to the best of our knowledge. However, in this research, only the remission period (a subset of the complete data) of the Verbal Fluency task is applied. In this thesis, the fullest extent of the verbal fluency test is practiced, and different state of the art networks are applied to the data available. Hence, with this thesis, the research's extent has become more expansive, and new methods are tested for the literature.

1.4 The Outline of the Thesis

Chapter 1 introduces the topic, motivation, and scope of the thesis.

Chapter 2 summarizes the prior studies related to the topics in the thesis. Firstly, the chapter summarizes previous works on the classification of psychiatric disorders using fNIRS and then refers to the other measurements and methods used to classify the BD. It also mentions the previous work that is studied by our research group to classify the BD.

Chapter 3 explains the detailed theoretical background that is needed to understand the thesis. The chapter starts with the theory behind the medical imaging method fNIRS and BD, a psychiatric disorder that is to be classified in the thesis's scope. Afterward, it explains the deep neural networks starting with the most fundamental parts of them. Finally, it explains the state of the art deep learning methods used for TSC to clarify its implementation over fNIRS data.

Chapter 4 summarizes and reviews the specific fNIRS data that is used as input the

neural networks implemented in this thesis.

Chapter 5 describes the specific architectures built for the classification of bipolar disease using fNIRS. It explains the training procedure and presents the training curves over training processes. Eventually, the chapter evaluates each model over k-fold validation through accuracy percentages. Finally, the neural networks are tested through 6 different test splits.

Chapter 6 concludes the thesis by summarizing the study and results.

CHAPTER 2

LITERATURE REVIEW

2.1 Psychiatric Disorder Classification Using fNIRS

Diagnostic classification of psychiatric disorders using fNIRS has recently become a preeminent strategy. As such, promising accuracy results are reported as in a range between 69% - 88.3. In 2010, near-infrared spectroscopy measurements were used to distinguish schizophrenia patients from healthy subjects using sixty schizophrenia patients with age and gender-matched control groups. Measurements were held during the Verbal Fluency Test, and patients were classified with an accuracy result of 88.3% [3]. Afterward, fNIRS were used to reach out to the diagnosis of children with attention-deficit hyperactive disorder (ADHD) in 2015. Even though an objective biomarker for that study was not established before the advancements of fNIRS technology, they achieved a classification with an area under the curve value of 85% [4]. In 2017, Yasumura et al. used machine learning to recognize the same disease (ADHD) on fNIRS measurement associated with the data collected from 108 children with ADHD and 108 typically developing (TD) children. Following the study, they achieved 86.25% discrimination accuracy between subjects [5]. Lastly, in 2019, Dadgostar et al. proposed a similar machine learning method with SVM was used to classify schizophrenia using 16 channels of fNIRS. In this research, a genetic algorithm (GA) was also applied for some channels to recreate fNIRS signals for signal arrangement. As a result, they obtained 87% classification precision for the recognition of schizophrenia on healthy subjects. After all, among those to our knowledge, the study carried out by our research group was the only study on bipolar subjects using deep learning, which has used fNIRS responses as the input. The highest accuracy channel was found as 69% (Evgin et al. 2019) [2]. All in all, these may well

suggest that studies that are recently using the fNIRS with deep learning can reach remarkable accuracy results as close to the actual diagnoses as possible.

2.2 Classification Methods For Bipolar Disorder

Even though this thesis's scope is to use fNIRS to diagnose BD, some other measurement techniques were used to classify BD. In 2015, [6] Functional Magnetic Resonance Imaging is used for the discrimination of BD. In the research they collect data from 21 bipolar disease (BD) patients, 25 major depressive disorder (MDD) patients and 23 healthy control subjects in order to specify differences due to the lack of indicators between BD and MDD. Despite the difficulties in identifying features, accuracy of 92.1% was achieved owing to selected discriminating features. Two prominent outcomes came to light: A remarkable role in differentiation between BD and MDD may be generated by frontal gyrus. Functional information prevails anatomical information by a wide margin while classifying these two disorders. The other research [7] is motivated by the idea that clinical observations lack competence. Therefore, diagnostic markers should distinguish between patients with BD and healthy individuals/individuals at familial risk for BD. Thus, the researchers used pattern classification with task-based functional magnetic resonance imaging (fMRI) to test their predictive value in discerning patients from others above. Eventually, results of patients with BD were as follows: accuracy of 83.5%, sensitivity of 84.6% and specificity of 92.3% compared to unrelated healthy individuals; accuracy of 73.1%, the sensitivity of 53.9% and specificity of 94.5% compared to their relatives with MDD; the accuracy of 81.8%, the sensitivity of 72.7% and specificity of 90.9% compared to their healthy relatives. As a result, assessing whole-brain pattern analysis of task-based working memory fMRI data would be useful in arriving an accurate individual classification, which may help clinicians in determining the true clinical uncertainty of BD. The last work [8] in this field is dedicated itself to building more coherent clinical phenotypes of BD by using machine learning and collecting data from PubMed, Embase and Web of Science, finding 757 abstracts and including 51 studies in the review. In order to differentiate the diagnosis of BD from other psychiatric disorders or healthy controls, multi-level biological data has been used in several studies. Conse-

quently, techniques resulted from using machine learning may provide clinicians with key information in the areas of diagnosis, individual care and prognosis orientation.

CHAPTER 3

THEORITICAL BACKGROUND

3.1 Functional Near Infrared Spectroscopy

Biomedical imaging is an essential method for the diagnosis of most diseases. Functional Near-Infrared Spectroscopy (fNIRS) is an imaging technique that is currently increasing in popularity. fNIRS measures the hemodynamic responses of the blood cells in the brain using near-infrared lights in the wavelength range of 700 to 900 nanometers [9, 10]. Infrared lasers having two distinct wavelengths are emitted to tissues using emitting sources. These sources irradiate the near-infrared light to the subject's scalp, and the reflected beams are collected using photodetectors (transceivers). The aforementioned emitting sources and photodetectors are the main parts of an fNIRS system. Some photons are absorbed and scattered by the tissue, and some can reach the photodetectors after following an arc-shaped path. Illustration of this process can be observed from the Figure 3.1.

Oxy and deoxyhemoglobin are both main absorbers of near-infrared lights, but they have different absorption spectrums. For each of the oxy and deoxyhemoglobin sensitivity levels, two lights with different wavelengths are emitted to the subject. The corresponding hemodynamic response can be acquired using a modified version of the Beer-Lambert Law. [12]

When compared to other imaging methods, fNIRS has both more advantageous and weak features. The time resolution of the sampled data is relatively strong compared to other imaging techniques. In addition to this, FNIRS is quite feasible as it uses near-infrared lights which is safe for the subjects, and its energy requirement is not as much as other imaging techniques. Also, fNIRS is more advantageous when applied

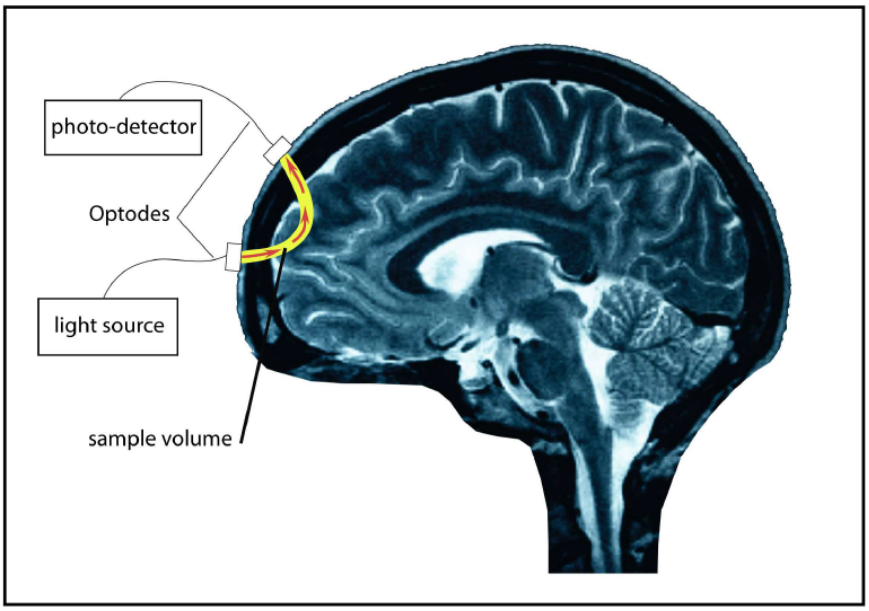

Figure 3.1: Beam emitting and absorbing using sources and detectors [11]

to more vulnerable subjects such as children, psychiatric populations, and even primates [13]. However, imaging depth is not as deep as other techniques since fNIRS is a non-invasive method and light is attenuated over the subjects skin and skull.

3.2 Bipolar Disorder

The illness that will be classified throughout in is called Bipolar Disorder (BD) which is a mental and chronic disorder known as manic-depressive illness [14]. The illness is characterized as at least one lifetime manic or mixed episode, and the emotional mood of the subject can change unusually. The early diagnosis and treatment of the disorder are essential since there is a possibility that the illness can affect the social life of the patient or, more importantly, the illness can lead the patient to suicide. The patient experiences two different states of the illness which are mania and depression. When the patient is in a mania state of the illness, the patient experiences positive mood changes such as feeling self-confident, energetic, and excited. In the depressive state of the illness, the patient experiences the opposite of the mentioned mood changes.

BD is a disease that mostly affects young adults and results in significant disabilities. In addition to the observed clinical symptoms, it is also known that there are some

results such as cognitive disorders. Lose of verbal fluency is one of these cognitive disorders. Although BD is known as an intermittent disease that results in mood episodes, recent discoveries point out that patients suffer from different kinds of malfunctions when they are in the normal state of the illness. This issue is vital because the anomalies detected when the patients are not in a mood episode offer trait markers that can be used to identify the disorder's neural biology and etiology. Although fNIRS has not been used to classify BD patients and control subjects, deep learning has already been used successfully to identify several diseases, as mentioned in chapter 2. The hypothesis that is aimed to prove in this thesis is classifying patients with BD with a control group that is matched by their genders and ages by training several deep learning architectures with fNIRS measurements.

3.3 Time Series

Time series are one-dimensionally ordered sets of values found in almost every domain in nature. Classification of time series is labeling these series according to different features that data has. Recently, deep learning is used for training classifiers using some training data and for validating these classifiers on data that has not been seen before. The details of the classification of the time series with deep learning will be explained in the following chapters. Time-Series' are categorized into two different categories:

- Univariate Time Series: These time series contain single scalar observations sampled through time. These samples are taken between equal time differences. A single channel digital audio signal can be an example of this time series (Figure 3.2).

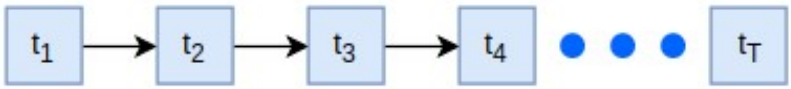

Figure 3.2: Univariate time series

- Multivariate Time Series: These types of time series contain M channels of univariate time series. Our fNIRS data corresponds to this category (Figure 3.3).

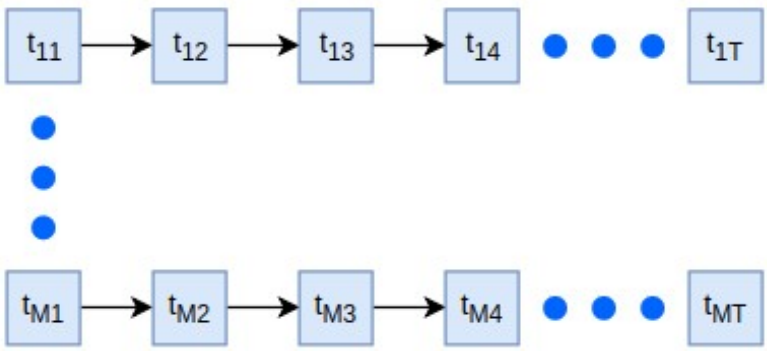

Figure 3.3: Multivariate time series

fNIRS are multivariate time series containing M number of channels. M is the number of the sources placed on the subject's scalp. In this thesis, these time series will be used to classify BD by training the neural networks that are going to be mentioned in the following sections.

3.4 Artificial Neural Networks

In this section, Artificial Neural Networks (ANN), ANN components, and the theory behind the classification using ANN is introduced and explained. Although the classification of time series data can be established using several methods such as conventional signal processing and many other statistical methods, deep learning and machine learning have been the most successful method in recent years. Its popularity is still increasing exponentially. ANNs are information processors that are one of the main tools and the most fundamental part used in machine and deep learning. These systems are inspired and constructed from human biological brains. From simple to more complex, the system consists of interconnected layers, and these layers are composed of some units called nodes. The nodes also resemble neurons in a bi-

ological brain. Nodes in every layer are connected to the nodes in the adjacent layer with so-called weights. Using these weights, information coming from the input accumulates layer by layer. ANN learns by using this information and updating these mentioned weights. These weights can be seen as synapses in biological neurons. A simple neuron structure can be seen as in Figure 3.4

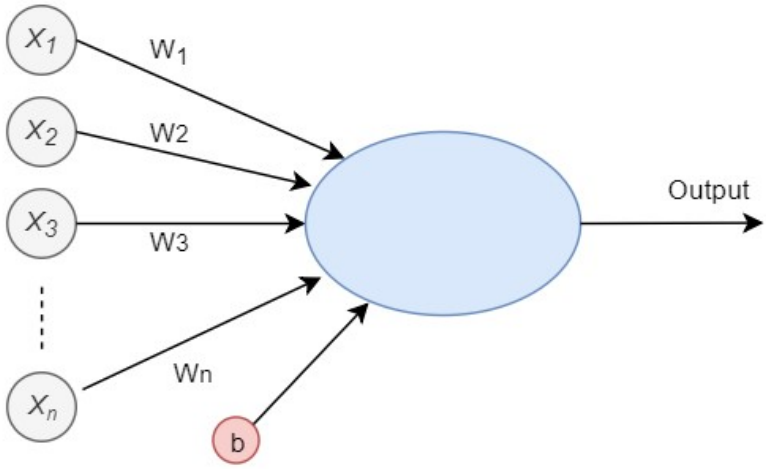

Figure 3.4: Single neuron model for an ANN

Each neuron sums the received information using the weights assigned to according input. Each input component $x \in X^{LAYER}$ is fed to the neuron using its own weight and these inputs are executed using the Equation 3.1

$$Output = \sum_{n} (w_n * x_n + bias) \tag{3.1}$$

An ANN generally consists of one input layer and one or more than one hidden layer. Each layer consists of predetermined amount of neurons, and the input layer consists of several neurons equal to the dimensions of the input fed to the network. A single layer can be visualized as it can be seen in Figure 3.5

Individual weights w generates a weight matrix W of a single layer and the bias vector b forms the parameters of that layer. The matrix representation of ANN's can be seen in Figure 3.6.

After the weighted sums are calculated, and output is obtained, this output must be scaled using an activation function. Moreover, classification problems are not al-

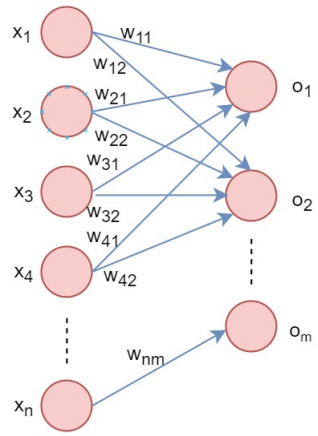

Figure 3.5: Single layer of an MLP

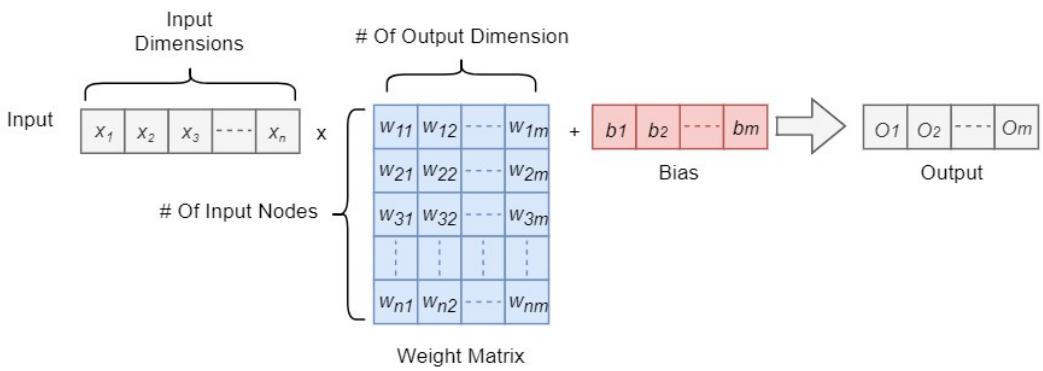

Figure 3.6: Matrix representation of a single layer of an ANN

ways linear classification problems, and non-linear classification boundaries should be added to the model. And these activation functions are mainly used to add this non-linearity to the model. Also, they are used to overcome vanishing gradient problems. Most commonly used activation functions are exemplified below.

- Step Function: The most straightforward activation function is a step function. By setting a threshold value, the neuron is activated accordingly. However, the step function is not preferred in recent studies in machine learning and deep learning since it is not capable of classifying not linearly separable data. When the data is linearly separable step functions can be used for the sake of

simplicity. (Equation 3.2) (Figure 3.7)

$$Step(x) = \begin{cases} 1 & \text{if } x >= 0 \\ 0 & \text{if } x < 0 \end{cases} \tag{3.2}$$

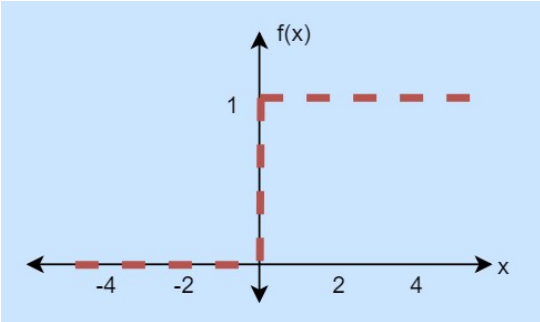

Figure 3.7: Step activation function

- Rectified Linear Unit (RELU): RELU is an activation function proposed by Nair and Hinton in 2010 [15]. Although the RELU activation function (Equation 3.3) seems to be linear in the positive axis, it, in fact, is a nonlinear activation function. The RELU activation function helps to prevent the vanishing gradient and saturation problems. Moreover, RELU gets activated sparsely, and this is desirable in most cases (Figure 3.8).

$$Relu(x) = max(0, x) \tag{3.3}$$

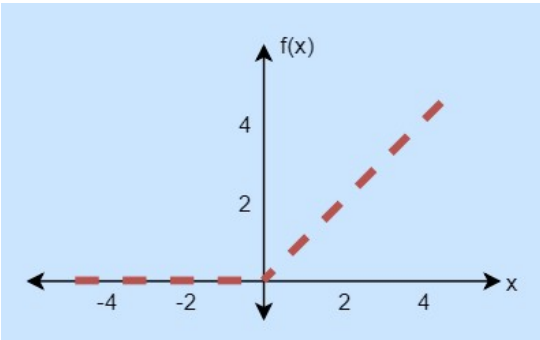

Figure 3.8: Relu activation function

- Sigmoid Activation Function: Sigmoid function (Equation 3.4) is the smoothened version of the step functions. Hence, the sigmoid function is capable of solving not linearly separable classification problems. Since this function bounds between 0 and 1, the obtained classification result involves the information about how 'confident' this classification is. Sigmoid functions are also known as logistic functions: Although the vanishing gradient problem still exists, the method is widely utilized in solving machine and deep learning problems.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.4}$$

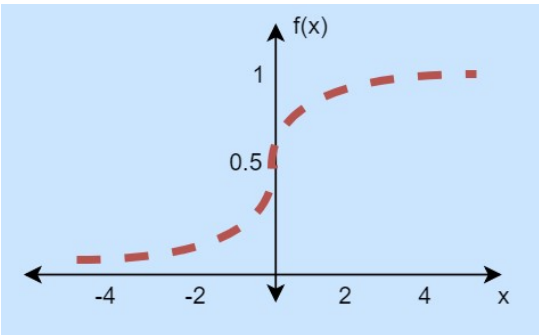

Figure 3.9: Sigmoid activation function

In order to evaluate the performance of the network on data modeling, loss functions are used. Loss functions are sometimes referred to as cost functions. Weight matrix W and bias vector b are optimized using this loss function for all training samples in the training set. Output Y vector calculated from input vector X is used with the ground truth $\tilde{Y}$. The aim of the algorithm is to minimize the loss $L(W, B)$ value. If the loss $L(W, B) \approx 0$ for all training samples, this means the algorithm is performing successfully. Generic loss function can be calculated as follows, where n is the nth training sample (Equation 3.5).

$$L(W, B) = \frac{1}{n} \sum_{n}^{N} L(W, B; y_n, \tilde{y}_n) \tag{3.5}$$

Some other widely utilized loss functions are

- Mean Squared Error

$$L(y_n, \tilde{y}_n) = \frac{1}{n}\sum_{n}^{N}(y_n - \tilde{y}_n)^2 \tag{3.6}$$

- Root Mean Squared Error

$$L(y_n, \tilde{y}_n) = \sqrt{\frac{1}{n}\sum_{n}^{N}(y_n - \tilde{y}_n)^2} \tag{3.7}$$

- Mean Absolute Error

$$L(y_n, \tilde{y}_n) = \frac{1}{n}\sum_{n}^{N}|y_n - \tilde{y}_n| \tag{3.8}$$

- Cross Entropy Loss

$$L(y_n, \tilde{y}_n) = \frac{1}{n}\sum_{n}^{N} y_n * \log \tilde{y}_n \tag{3.9}$$

An artificial neural network enhances its performance by updating the aforementioned weight matrices. Each layer has its own weight matrices which have to be updated at each iteration accordingly. These updates should be done according to the loss functions. The forward pass of a neural network can be defined as the calculation process of the output using the input data as well as the hidden layers. The output of the forward pass is the input of the loss function. With X being the input layer of the artificial neural network $Z^{(2)}$, the output of the first layer can be formulated as:

$$Z^{(2)} = W^{(1)}x + b^{(1)} \tag{3.10}$$

Activated output $a^{(2)}$ can be formulated as:

$$a^{(2)} = f_{activation}(Z^{(2)}) \tag{3.11}$$

Using the activated output of a layer, the output and the activated output of nth layer can be computed as:

$$Z^{(n)} = W^{(n-1)}a^{(n-1)} + b^{(n-1)} \tag{3.12}$$

$$a^{(n)} = f_{activation}(Z^{(n)}) \tag{3.13}$$

The final output Y and the loss calculated from the forward pass can be formulated as:

$$Y = f_{activation}(Z^{(n)}) \tag{3.14}$$

$$L = Loss(Y, \tilde{Y}) \tag{3.15}$$

The loss function is the numeric output that calculates the update rate and the update direction of the weight matrices. For each iteration, weights of each layer will be updated using an update rate α, which is also called the learning rate. This update can be written as

$$W^{(l)} = W^{(l)} - \alpha * \frac{\delta L}{\delta W^{(L)}} \tag{3.16}$$

For each layer, the gradient with respect to the weights needs to optimize the weights' change rates. This algorithm is called stochastic gradient descent. So by applying the chain rule, the gradient with respect to the last layer can be written as

$$\frac{\delta L}{\delta W^{(n)}} = \frac{\delta L}{\delta \alpha^{(n)}} * \frac{\delta \alpha^{(}n)}{\delta z^{(n)}} * \frac{\delta z^{(n)}}{\delta W^{(n)}} \tag{3.17}$$

In order to get deeper in layers chain rule is applied multiple times

$$\frac{\delta L}{\delta W^{(n)}} = \frac{\delta L}{\delta \alpha^{(n)}} * \frac{\delta \alpha^{(n)}}{\delta z^{(n)}} * \frac{\delta z^{(n)}}{\delta \alpha^{(n-1)}} * \frac{\delta \alpha^{(n-1)}}{\delta z^{(n-1)}} ... * \frac{\delta z^{(1)}}{\delta W^{(1)}} \tag{3.18}$$

The same rule is also applied to the bias term and this can be written as

$$b^{(l)} = b^{(l)} - \alpha * \frac{\delta L}{\delta b^{(l)}} \tag{3.19}$$

This process is called backpropagation [16]. Each forward pass following by a backward pass is called an iteration. All the iterations that iterate over a training dataset are called an epoch. The epoch number is one of the most critical tunable parameters of a neural network and must be tuned carefully to avoid overfitting.

Besides gradient descent, there are plenty of optimizers that are commonly used in ANN and deep learning. Some of them that are used in the scope of this thesis is

presented as follows:

- Momentum Optimizer: Momentum is a term that can be used to upgrade SGD by updating the weights using previous update rates to use the momentum of the gradient of the curvature. As the gradient gets larger values, learning gets faster. Update rate can be written as

$$W^{(l)} = W^{(l)} - \upsilon(t) \tag{3.20}$$

$$\upsilon(t) = \gamma * \upsilon(t-1) + \alpha * \frac{\delta L}{\delta W^{(l)}} \tag{3.21}$$

- Adaptive Moment Estimation (ADAM) Optimizer: ADAM [17] is an optimizer that uses first and second-order momentums that are shown in equations (3.22) and (3.23). ADAM's main use is not to miss the global minimum by jumping over because of the large momentum.

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{3.22}$$

$$\hat{\upsilon}_t = \frac{\upsilon_t}{1-\beta_2^t} \tag{3.23}$$

  Parameter optimization is done as it is shown below:

$$\theta_{t+1} = \theta_t - \frac{\mu}{\sqrt{\hat{\upsilon}_t} + \epsilon}\hat{m}_t \tag{3.24}$$

The structure explained in this section is also referred to as Multilayer perceptrons [18] (MLP) and Fully Connected Neural Networks. Although the terms are being used reciprocal to each other, in the proceeding chapters, the term MLP is used for the reference.

There are few methods to prevent the overfitting problem. However, in the scope of this thesis, the primary technique that is used to prevent overfitting is using dropouts [19]. Dropout randomly wipes off some neurons with a particular percentage parameter, and this enables the network not to rely on individual neuron connections. The dropout should not be used in the network's final layers to have the ability of fixing the errors resulting from the dropout layers.

Finally, Global Average Pooling layer is an important context in this thesis' scope. Global Average pooling layers are used to apply averaging operation over feature maps [20]. They generally are used in the end of the convolutional layers for summarizing temporal informations.

3.5 Deep Learning

Deep learning is a process that requires hardware highly capable of handling matrix and vector multiplications. With the advances in hardware technologies, using graphical processing units (GPUs), it became possible to implement high scale matrix and vector multiplications in parallel. Without the GPUs, these implementations had to be done in serial using central processing units (CPUs). GPU's parallelism increases the processing speed of deep learning training and enables real-world problems to be solved using deep learning. Nowadays, even the most simple problems can be approached using deep learning. MLPs are the simplest ways of using deep learning. In this part, the methods used in this thesis are Convolutional Neural Networks (CNN) and some modified convolutional network architectures: Residual Neural Networks (ResNet) and Encoder Networks described. Although these networks are initially designed for computer vision problems, they are modified for being used in time series problems [21]. In this thesis's scope, fNIRS data, which is a multivariate time series, is classified for the classification of BD using deep learning. A simple deep learning framework for time series classification can be examined in Figure 3.10.

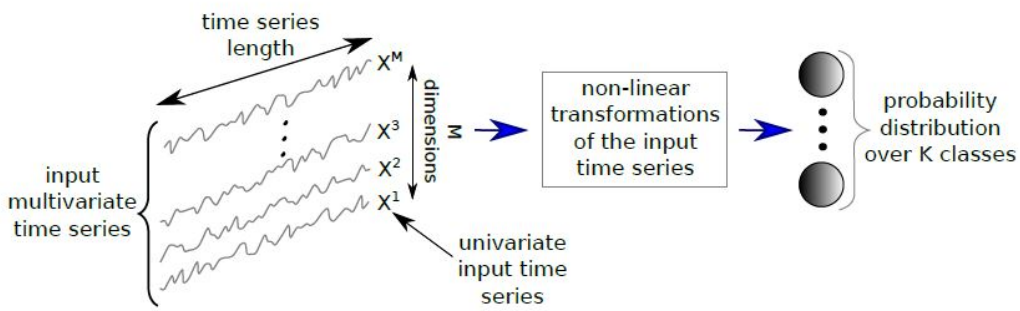

Figure 3.10: Deep learning framework for time series classification [21]

3.5.1 Convolutional Neural Networks

CNNs [22] are deep learning architectures that take n dimensional inputs and assigns different importance levels to different parts and features of these inputs. For the case of this thesis topic, the inputs are one-dimensional multiple channeled inputs. However, CNN's are commonly taking place in problems, including 2-dimensional inputs, which are images. Input images are typically made of three independent channels, abbreviated as RGB, where R stands for the red channel, G stands for the green, and B for the blue. Each channel can be conceptualized as a filter. For convolutional neural networks, a layer consists of determining width, height, and depth. A typical neural network can be visualized as shown in Figure 3.11.

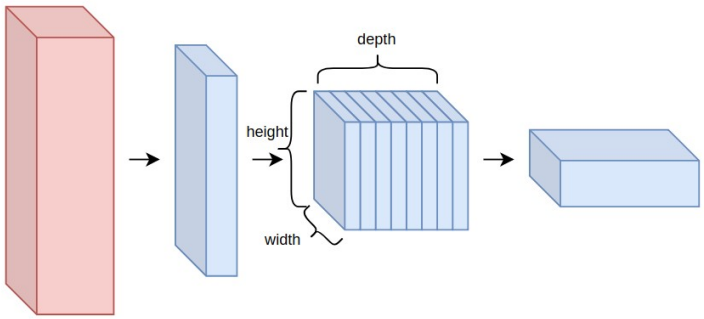

Figure 3.11: Visual representation of a CNN and the height, width and depth of a single convolutional layer

Depth of a convolutional neural network is defined as the number of filters in a distinct convolutional layer. As the number of filters on a layer increases, the layer gets more capable of extracting features; however, the training process on that layer gets slower and more prone to overfitting. The reason behind that is the current input might not be containing as many features as it is predicted to have. As the layers get more in-depth, the neural network will be able to learn more complex features. These layers constitute a hierarchical structure for different data. As an example for this hierarchy, for a simple face recognition or classification task, the first layers learn simple features like edges, curves, etc., and the higher layers learn complex natural structures such as the eyes and the noses. Suppose the activations of the filters in different layers of a pre-trained neural network are visualized. In that case, the hierarchical structure can

be examined as it can be seen in Figure 3.12.

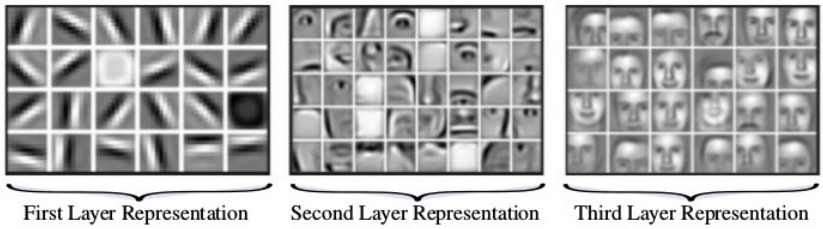

Figure 3.12: Hierarchical structure of the layers of an Convolutional Neural Network [23]

CNN's dimensionality can be manipulated arbitrarily. As it is studied in this thesis, single-dimensional convolutional neural networks are derived and used to extract features from one-dimensional data in many applications for problems with fewer dimensions. A single convolution operation by a single filter for two dimensional image can be seen in Figure 3.13.

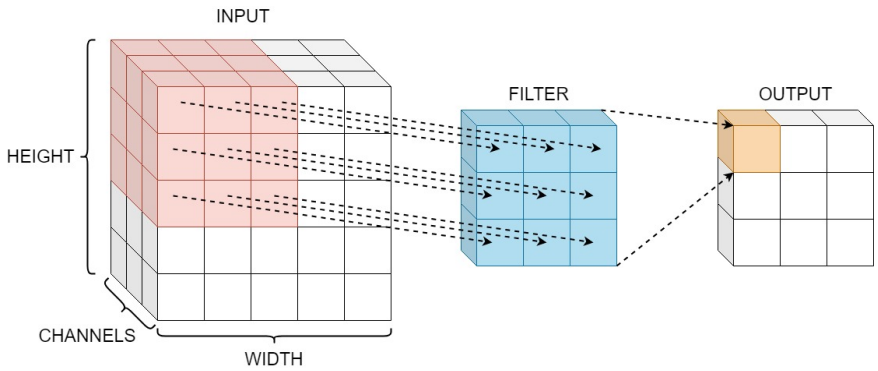

Figure 3.13: Convolution operation for a 2 dimensional CNN

However, in this thesis, the convolutional neural networks are modified for time series classification. So the convolutional filter is modified to have a single dimension for the convolutional operation (Figure 3.14)

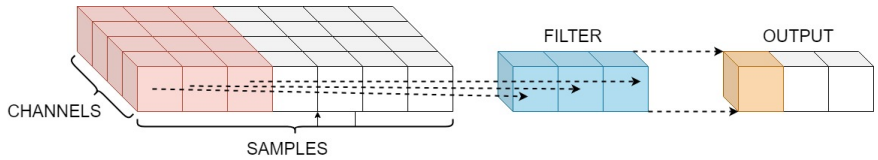

Figure 3.14: Convolution operation for a one-dimensional CNN

After CNN structure proves its success on computer vision problems, in 2017, deep learning started to be used in time series analysis [24]. Although the dimensionality reduces, it is hard to visualize one-dimensional convolutional neural networks. For visualization problems, multiple channels in the input refers to the RGB channels of a camera frame. The convolution process occurs for different channels in parallel and they are added to the next layer. However, for one-dimensional problems, the channels are temporal inputs sampled simultaneously, and the convolution occurs through these channels. In 2016, Fully Convolutional Neural Networks were suggested to use for the time series classification along with MLP and ResNet in the same research [1]. This research has become a strong baseline for time series classification with deep learning. In that research convolutional layers are used without fully connected layers in the end of the network. Instead of fully connected layers, Global Average Pooling (GAP) layer is used for averaging the time dimensions.

3.5.2 Residual Neural Networks (ResNet)

Convolutional neural networks have proven to be successful by obtaining outstanding results in many different classification problems. As the convolutional neural networks achieve success by stacking more layers and by constructing "very deep" models, it is started to be researched that if the learning is capable of increasing proportionally by adding more layers. The problems of vanishing and exploding gradients are solved by different methods such as normalized initialization and intermediate normalization layers; however, when the depth of network reaches to a significant amount, the accuracy of that network gets saturated, and after a while, it starts to decrease sharply. The reason of this decrease does not stem from the overfitting since the training accuracy also decreases after that significant point. In order to construct deeper networks, the problem mentioned above is addressed by the networks called Residual Neural Networks (ResNet). ResNet introduces some shortcut connections called residual blocks (Figure 3.15). These blocks connect between previous layers using identity mappings that do not have any parameters. These layers connect layers that will use the shortcut by adding previous ones to ahead ones. Moreover, the network's complexity does not increase since the shortcut layers do not add any learnable parameters to the network.

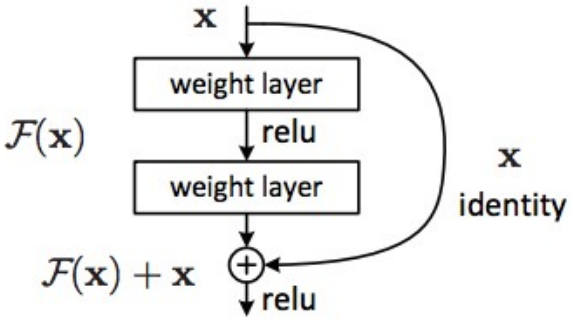

Figure 3.15: Representation of a shortcut block in a ResNet [25].

As the input gets convolved through convolutional layers, the dimensions of the tensor get lower. However, the shortcut connections are trying to add these previous layers to the convolved output. To make this possible, the identity mapping is multiplied with a linear projection matrix to expand the channels of shortcuts. These shortcuts are added to different intermediate parts of the networks; hence obtaining deeper networks becomes possible. For time series classification, the most in-depth architecture proposed by Wang et al. [1] used an 11 layered network using residual connections. As it happens in convolutional neural networks, the two-dimensional filters are replaced with one-dimensional filters, and the convolution operation is applied. The architecture proposed by Wang et al. can be seen in Figure 3.16.

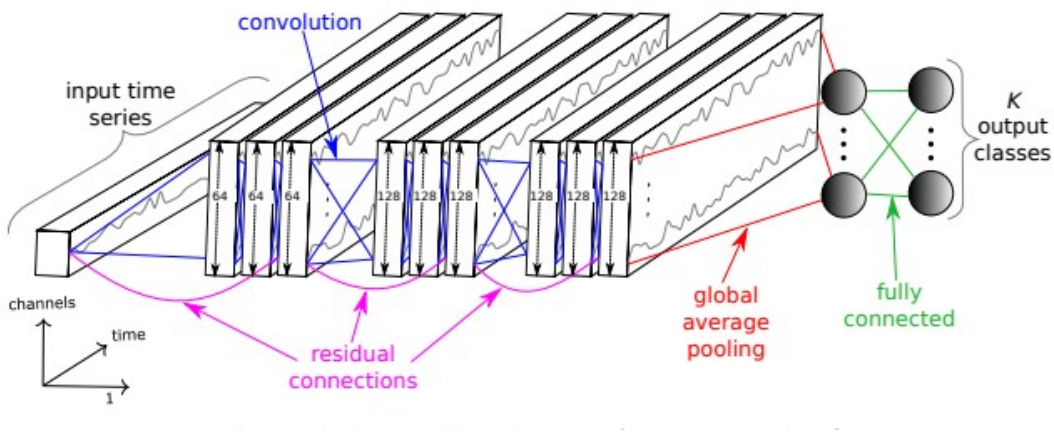

Figure 3.16: Time ResNet Architecture suggested by Wang et al. [1]

3.5.3 Encoder Network

Encoder networks architecture [26] is a neural network architecture similar to the CNN architecture. Like the CNNs, Encoder networks also uses convolutional layers to extract the temporal correlations on the time axis. In the first part of the network, the fully convolutional networks' outputs are fed to instance normalization layers[27]. Following the normalization, PRELU activation functions and dropout are applied to the output. Finally, outputs are applied to one-dimensional max-pooling layers. The summarization of the multidimensional information for multivariate time series is one of the most important parts of time series classification. CNNs use the Global Average Pooling method for this summarization. Encoder networks replace the Global Average Pooling layer with a mechanism called attention mechanism (Figure 3.17), which is proposed initially by Bahdanau et al. [28] for the second part of the network. This mechanism is also the main difference between the encoder networks from CNNs.

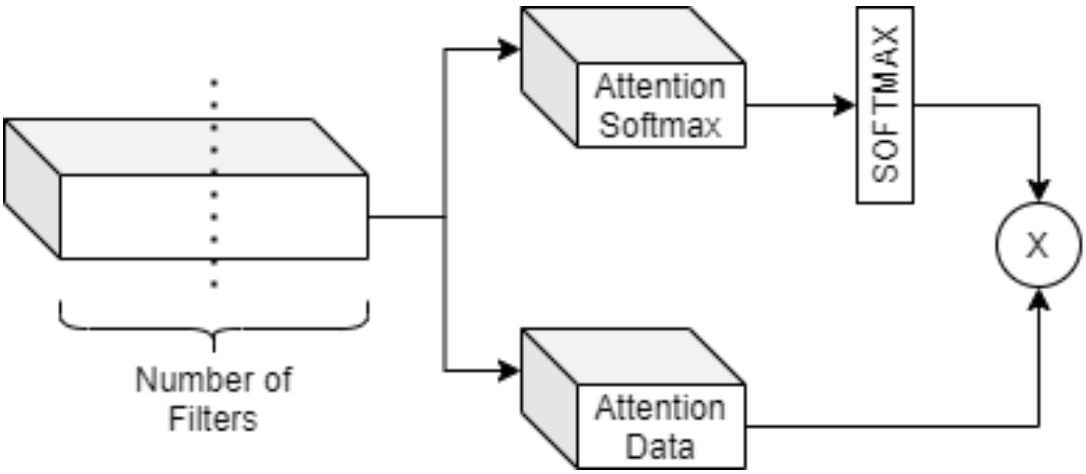

Figure 3.17: The attention mechanism

The output O of the filters after the last convolutional layer is splitted into the two parts as $O[:, \frac{n}{2}]$ and $O[\frac{n}{2}\ :]$. One part is applied to a Softmax function and treated as a timewise Softmax attention mechanism for the other part of the output of the filters and these parts are multiplied together as it can be examined in Equation (3.25).

$$AttentionOutput = O[:, \frac{n}{2}].Softmax(O[\frac{n}{2}, :]) \tag{3.25}$$

The resulting output is fed to fully connected layers for the classification output after

an instance normalization layer at the neural network's end. Although this method is initially proposed for the implementation of transfer learning for time series classification, training the network from scratch has also given quite successful results [21].

CHAPTER 4

DATASET INTRODUCTION

4.1 Data Review and Verbal Fluency Test

Neuroimaging data used in this thesis comprise cortical activity during a verbal fluency task acquired from subjects with BD and healthy control subjects. The original data was collected in a previous completed thesis in the field of adult psychiatry by Hoşgören [29]. Data was collected with the Hitachi ETG-4000 optical imaging system located at Ankara University Brain Research Center between 01.04.2014 and 30.07.2014. The distance between emitter and detector optodes are set to 3.0 centimeters and each channel is determined as the area between the emitter and detector. These placements are done with respect to EEG International 10-20 System (Figure 4.1).

Thanks to the filtering and noise-canceling features of the measurement system, besides the raw data, filtered and noise-free data can be used without any preprocessing process. Although the measurements have been taken from 82 subjects, 11 of them are marked as not usable. For five of them, the measurement system's noise cancellation feature has not been applied. For the other six subjects, the data had missing parts.

For each of the bipolar subjects, it is important to highlight that each subject is in the remission period of the disorder. Which means when the data is being collected, the patient does not experience any mania or depression behaviour.

The task used to measure verbal fluency was an adapted version of the Controlled Oral Word Associations Task [30]. The VF-task consisted of a 30 seconds pre-task

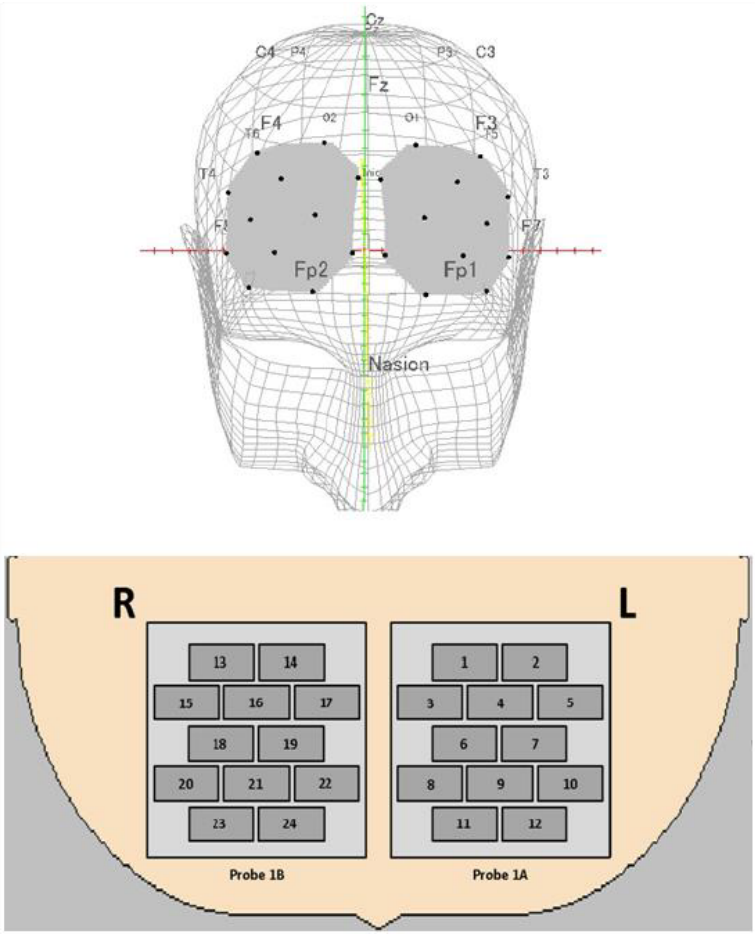

Figure 4.1: EEG International 10-20 System for optode placement

period and a 30 seconds task period and was presented as two blocks. In each block, three letters were sequentially presented and appeared for 10 seconds on a 21-inch screen placed 50 cm ahead of the participants' scalp. Participants were instructed to produce as many words as they can starting with the presented letter. Letters /s/,/a/,/k/ are used in the first, and /m/,/i/,/s/ are used in the second block. Since the participants responded verbally to the task, they were instructed to repeat the voice vowels constantly (/a/,/e/,/o/) during the baseline periods to control the effect of articulation on brain activity. Data has been collected from 24 channels for two wavelengths with a 100 milliseconds time resolution. For each subject, a measurement took 225.1 seconds. From each channel, 2251 samples are taken, and for all channels, 108048 samples are collected from a single subject.

The first 30 seconds of the task is named the pre-task period, and for these first 20

seconds, the subjects are at rest. The following two 30 seconds blocks are called the task time. After these blocks, the next 20 seconds are named post-task period, and the subjects are also at rest for these 20 seconds. The task time was kept unchanged for the second task period as 60 seconds, but the pre and post-task times are chosen as 25 and 30 seconds. The visualized task time structure can be seen in Figure 4.2.

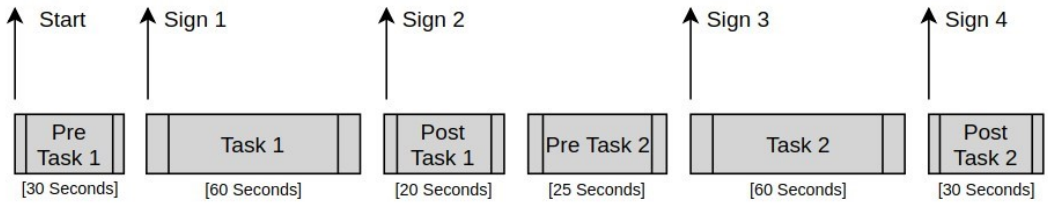

Figure 4.2: Time graph of the task applied to the subject

4.2 Outlier Detection and Dataset Summary

After the dataset constructed with 71 subjects, being worked with simple architectures, it is observed that the training curves are nearly impossible to prevent from overfitting, and binary classification could not be satisfied. After a few more trials with different architectures, the results revealed some outliers in the data. In order to determine the outliers in the data, All the subjects in the data are indexed with unique ids. In order to prevent overfitting, the test sets are chosen to be relatively small, and the subjects are trained and tested using random shuffles. After reaching a sufficient number of tests, the success rates for each subject are noted. When the results are investigated, it is observed that some samples are consistently labeled wrong and affects the training curves adversely.

A total of 79 random shuffles were conducted, and the subjects appeared in the test set 493 times. Each subject appeared 6.94 times in the test set on average. The worst resulting ten subjects with success rates under 37.5% percent are labeled as outliers and removed from the dataset. The resulting data set contains 61 subjects containing 28 bipolar and 33 control subjects.

To construct the dataset for the training, firstly, the 10 samples are separated from

the leading group to use them for testing the trained networks. These 10 samples are chosen randomly. However, to have the test set as balanced as possible, it is checked that there is a balance as close as possible to 5 bipolar and 5 control samples in each test set.

The remaining dataset with 51 subjects is tagged as a training + validation set. These samples are not separated as training and validation to apply k fold cross-validation.

In order to apply k fold cross-validation, the dataset is shuffled randomly first. After the shuffling, the dataset is split into k groups. For our case, the k is selected to be 5 in order to increase the number of training samples occurring in the training set. This configuration enables the network to use at least 40 samples in the training set, which is 66% of the complete dataset. For training, one of the groups is held back as the validation set, and the other groups are used on the training. The training data gets fitted on the model and evaluated on the validation group. After the completed training and evaluations, models are discarded for testing the new fold to prevent the training leakage.

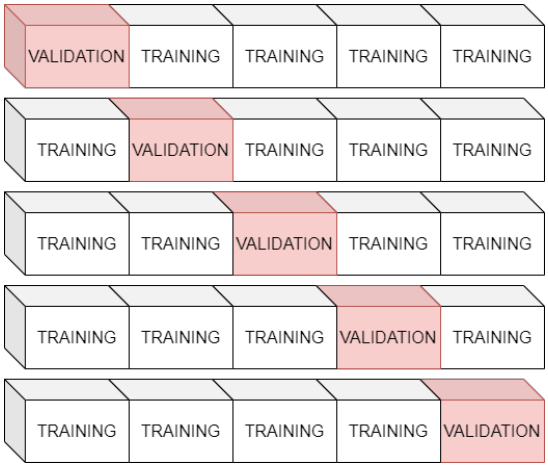

Figure 4.3: 3 Fold Cross Validation Method

For classification problems, each fold's representation is essential; for this purpose, the stratification process is used as much as possible. For example, if two classes have 50% percent of the data for a binary classification problem, it should be noted that each fold has to have a balanced number of subjects from each class. For the fNIRS data, it is impossible to have 50% percent in each fold; however, while shuffling the

data, it is considered to have the dataset as balanced as possible.

Finalized data have 51 shared training and validation subjects, and these subjects are divided into 4 groups of 10 subjects and 1 group of 11 subjects. The finalized test set contains 10 subjects equally separated by bipolar and control subjects. The training curves exhibited in the following chapter uses the mentioned training and validation data.

Since the test set needs to be representative of the whole population of the dataset, random sampling is not sufficient for choosing the test set. In order to solve this problem, it is decided that testing the whole dataset is vital for the evaluation of the models that are going to be mentioned. Hence, the process explained is repeated 6 times with different test sets. After a test set is tested with the best training fold found on the validation process, the models are discarded for the new test set to be tested.

CHAPTER 5

IMPLEMENTATION AND RESULTS

5.1 Methods

This section introduces the proposed architectures and the methods used to classify BD. The features are extracted using the proposed architectures, and the classification has been applied using these methods.

As it is mentioned earlier, because of the difficulty of sampling the subjects from the population as a test set, the complete dataset is separated into 6 partitions in order to test them individiually. Firstly, a single group is chosen from the population as a test set. For a more reliable training process with less data, five-fold cross-validation method is applied to the remaining 5 partitions to find the best resulting fold on the validation set as it is explained in chapter 4. Hyperparameter tuning for each model is carried out during the cross-validation process. Binary cross-entropy loss function is used for all models. After reaching a confident state in the validation process, the trained models are tested on the test set which is divided earlier. When testing for the first group is completed, the trained models are discarded and the process is repeated by the new training, validation and the test group until the testing of all 61 subjects is completed. Average classification accuracy of the 6 test cycles is recorded as the model classification accuracy. The schematic of each test can be seen from Figure 5.1.

Sensitivity (ratio of the true positive bipolar samples to all bipolar samples) and specificity (ratio of the true negative control samples to all control samples) of the models are calculated using the same method as well. The training curves presented in the following experiments are taken from the best resulting fold for all models. NVIDIA

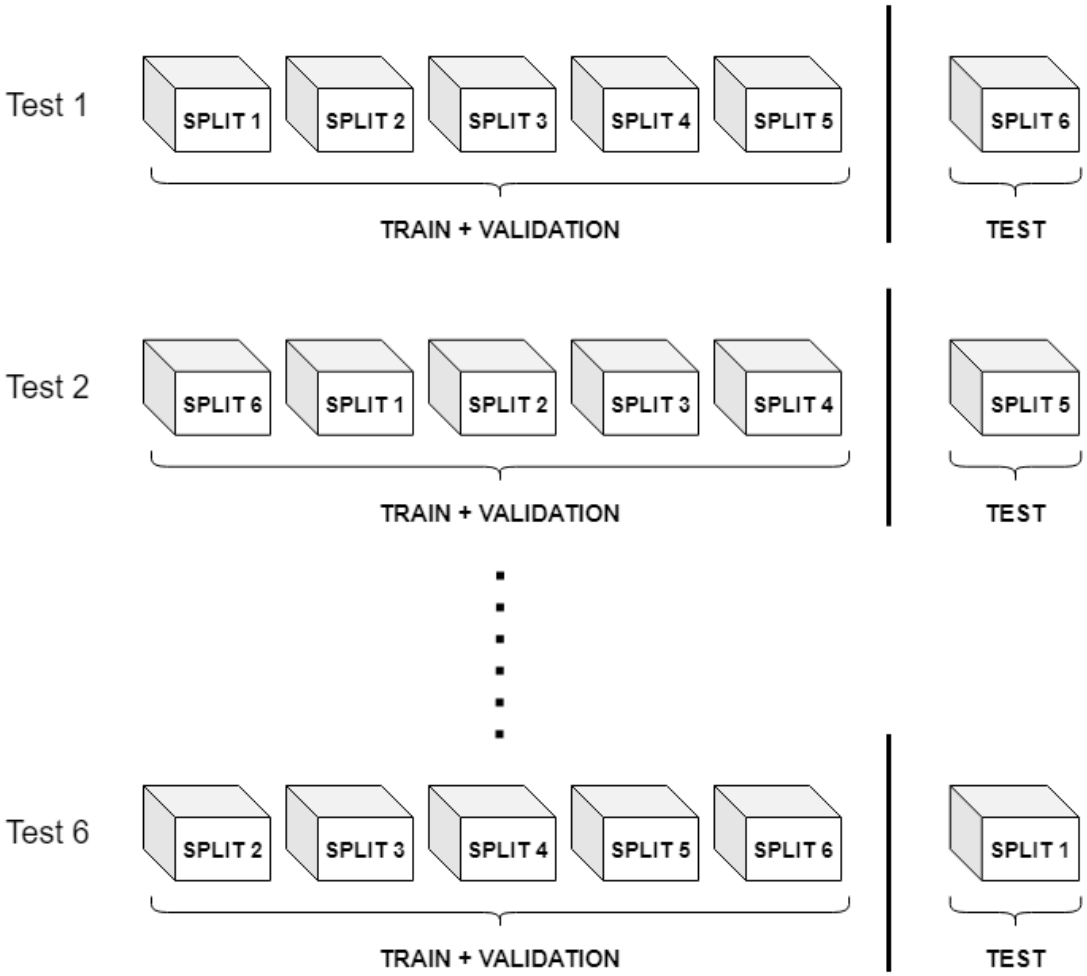

Figure 5.1: Training, validation and test diagram groups for each test

GTX 1050TI GPU is used for the training process. The code is implemented by using Tensorflow 2.1.0 [31] and Keras API 2.3.1 [32]. Since the data is few in terms of subjects, a balanced validation set has been constructed with ten subjects from the bipolar group and ten subjects from the control group. All trainings are repeated ten times, and shaded lines on the graphs illustrate individual graphs from tests. For all the networks that are being assessed, the same set is engaged for a fair comparison.

In the following subsections of the chapter, the main goal is to investigate training and validation graphs to suggest methods on the best resulting fold. Hyperparameter tuning and finding the optimal structures are put into effect by practicing the technique mentioned above.

5.1.1 Multilayer Perceptrons

Before implementing more complex structures, the MLP network, which is also known as Fully Connected Networks (FCN), is selected to be the starting point of the research due to its simple implementation. Although the fNIRS data is a time structured one-dimensional data, it still necessitates being flattened over its channels to obtain a feasible input for an MLP network. Nevertheless, flattening operation over 24 channels results in an immense number of samples. The data possesses 2251 samples per 24 channels and two wavelengths (700 nanometers and 900 nanometers). For a channel, the network's input must have 2251 samples multiplied by 48 channels resulting in 108048 input points, which means several nodes in the first layer. This increase represents a vast number of parameters to train at the first layer of the neural network. As the parameter numbers increase in number, the operation becomes more intricate to train and hypertune the network.

MLP is not anticipated to perform better than architectures having convolutional layers due to the flattening operation mentioned. Although there are no exact ways to resolve how many nodes should there be in intermediate layers of the MLP, there are some researches [33] that suggests different ways for seeking the best number of nodes in a hidden layer. For the fNIRS data, the number of hidden layers and the neurons in each hidden layer is achieved by experimentation. Hence, the best performing MLP architecture is found to have three hidden layers. The first layer connected to the input has 32 nodes, and the two consecutive layers have 64 neurons. Having fewer hidden nodes in the first layer is to keep the number of trainable parameters in the first layer as low as possible to decrease the training time. The activation function is chosen to be the RELU after these layers. The final layer has two nodes for the classification. As an optimizer, the ADAM optimizer with first and second moments of 0.8 and 0.9 is used to optimize the network. The schematic of the summary of the network can be examined in Figure 5.2.

The following training and validation curves (Figures 5.3, 5.4, 5.5, 5.6) are collected using the aforementioned structure on a sample fold for 5 fold cross-validation. Since the data's temporal channeled structure is corrupted due to the flattening operation for the network, MLP is not expected to outperform other structures tested in the follow-

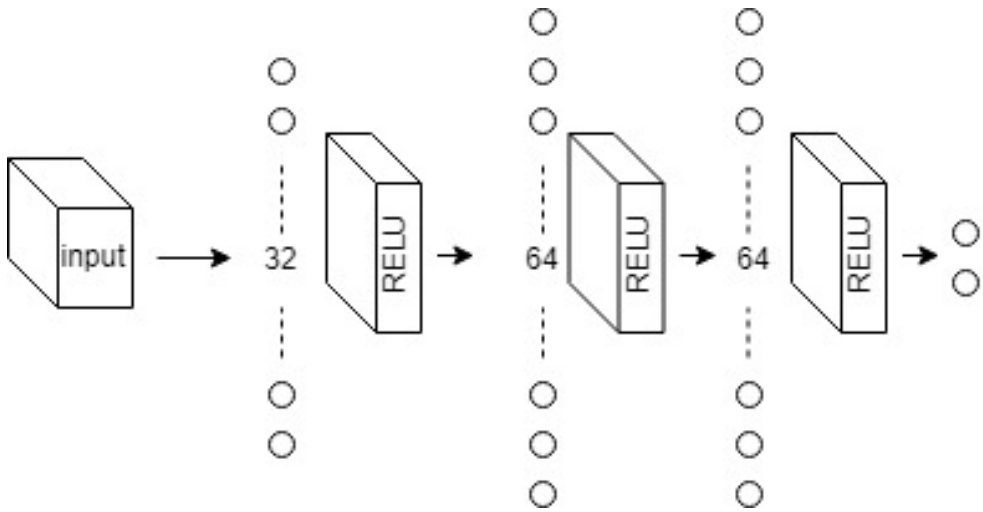

Figure 5.2: Summary of the Multilayer Perceptron model

ing sections for the fNIRS data. The shaded curves are taken from single repetition from all runs, and the red curve is the average of 10 repetitions. For all repetitions, the network is randomly initialized after discarding the previous network.

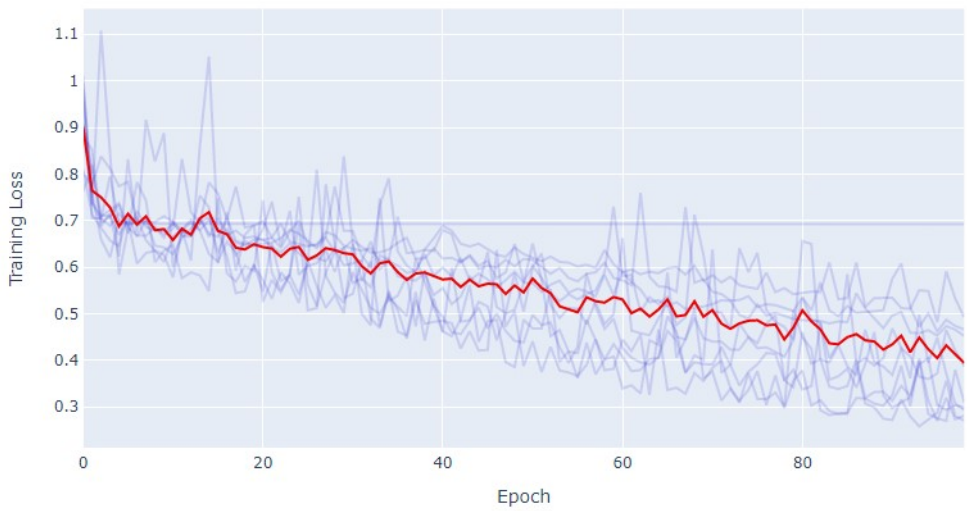

Figure 5.3: Training loss curve for the proposed MLP model

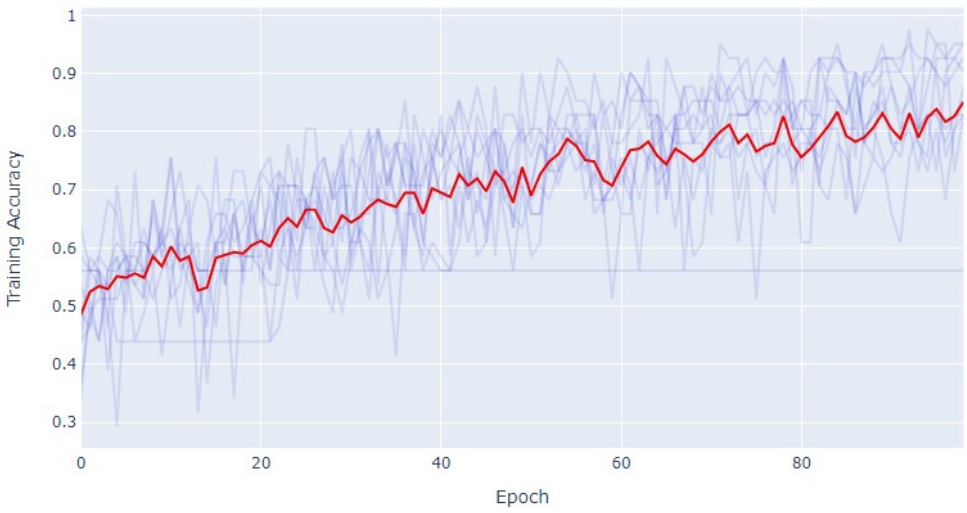

Figure 5.4: Training accuracy curve for the proposed MLP model

As the training curves are investigated, it is seen that the neural network can generalize the data on the training set, which also means the network is capable of learning the data with this method. Still, the training accuracy and the loss saturates at a point, and this result can be examined in training curves (Figures 5.3, 5.4). The reflection of this saturation is visible in the validation curves (Figure 5.5, 5.6) as the validation accuracy can not get larger than 70% percent before overfitting.

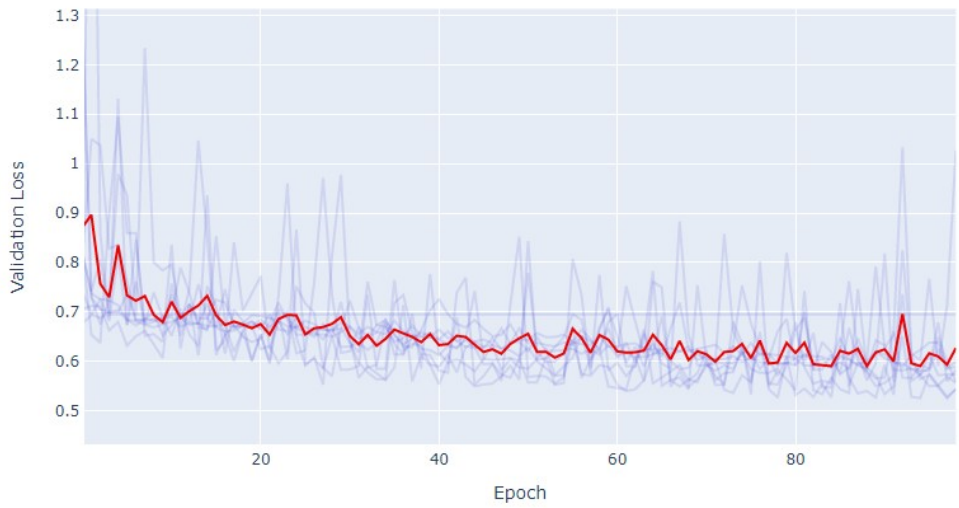

Figure 5.5: Validation loss curve for the proposed MLP model

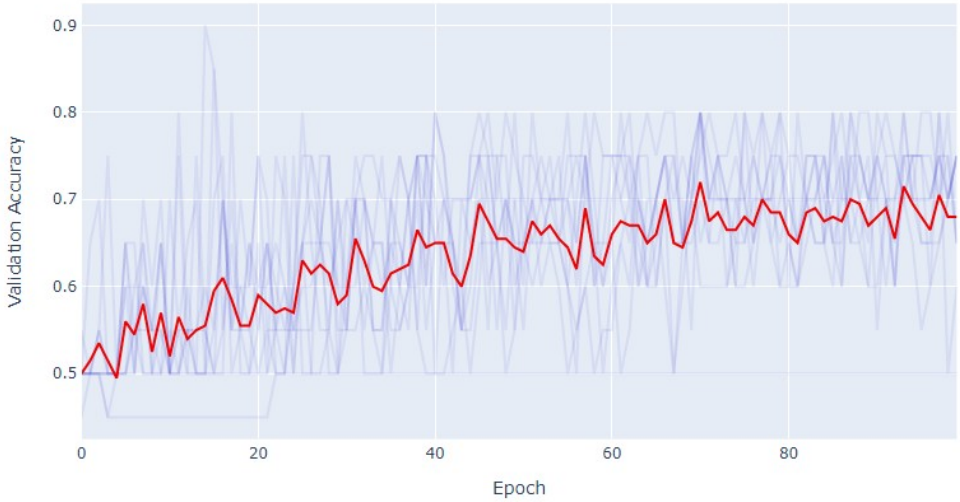

Figure 5.6: Validation accuracy curve for the proposed MLP model

As known, MLP has disadvantages since the network disregards the spatial information due to the flattening operation. Also, since all the nodes are connected to each other, the number of parameters increases to a large number, and training and testing operation becomes heavier. However, still obtaining reasonable accuracies on the validation set by using the most fundamental structure of artificial intelligence is a fair starting point for the research.

Although the given training curves only represent training for a single fold with a single test set, the tests are applied six times for different test splits after discarding the trained models for each split. Mean accuracy of 64.46% is obtained for six different test sets. Classification accuracy for each class and the overall accuracy result can be observed from Table 5.1

Table 5.1: Classification accuracy on each test split for the proposed MLP architecture

Split	1	2	3	4	5	6	μ	σ
Overall	67.09%	51.6%	64.3%	68%	61.55%	74%	64.42%	6.88
Bipolar	60%	52%	60.3%	50%	64%	82%	61.38%	10.43
Control	73%	50%	66%	80%	60.5%	66%	65.92%	9.41

As expected, a generalization of the channeled time series data using MLP is not a great success. However, before working with more complex structures based on convolutional layers, obtaining 64.46% percentage of testing accuracy is a successful starting point.

5.1.2 Convolutional Neural Networks

Due to its proven success on visual classification tasks, Convolutional Neural Networks (CNN) is started to be used in one-dimensional problems, and they give quite successful results. Hence, one-dimensional CNN is chosen to be the first proposed complex architecture for the fNIRS classification task. The channeled temporal structure of fNIRS data enables us to build one-dimensional multi-channeled CNN for multivariate time series data. The Fully Convolutional Neural Network model proposed by Wang et al. [1] for the TSC is taken as a baseline for the proposed CNN model. Since CNN's are the most reliable architecture for classification tasks, the work started by searching for an architecture that overfits the training data and giving an unsuccessful performance on the validation. The reason for exploring the model that overfits the data is essential to know whether the model is capable of generalizing the fNIRS data. This process is repeated for every model; however, just the result from CNN is exhibited for an example. After some parameter tuning, the following graphs have been obtained from the proposed model. As it can be seen in the Figures 5.7 and 5.8 the model is very successful in classifying the data on the training set. The accuracy of the training curve nearly reached the maximum, and the loss is decreasing sharply. The shaded curves are taken from individual runs, and the red curve is the average of 10 repetitions. All tests are repeated by starting the neural network by random initialization.

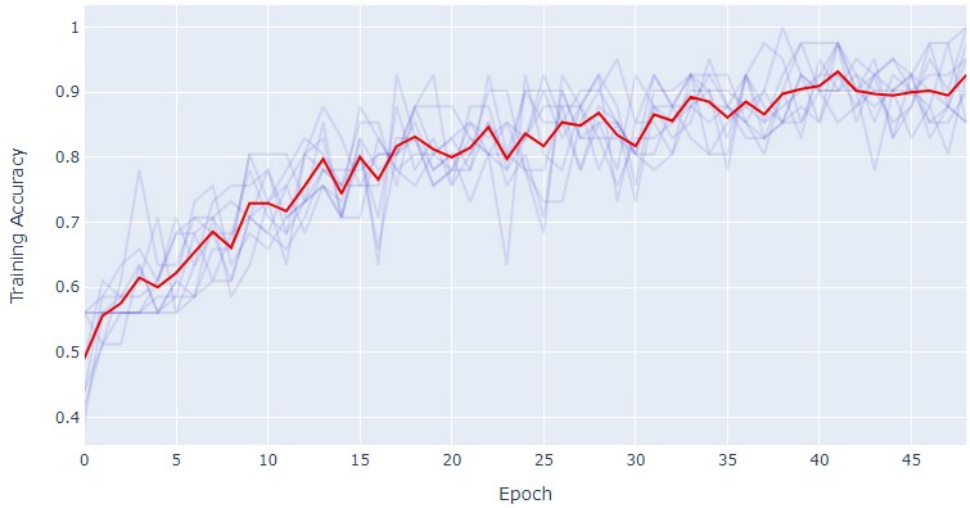

Figure 5.7: Training accuracy curve for overfitting CNN model

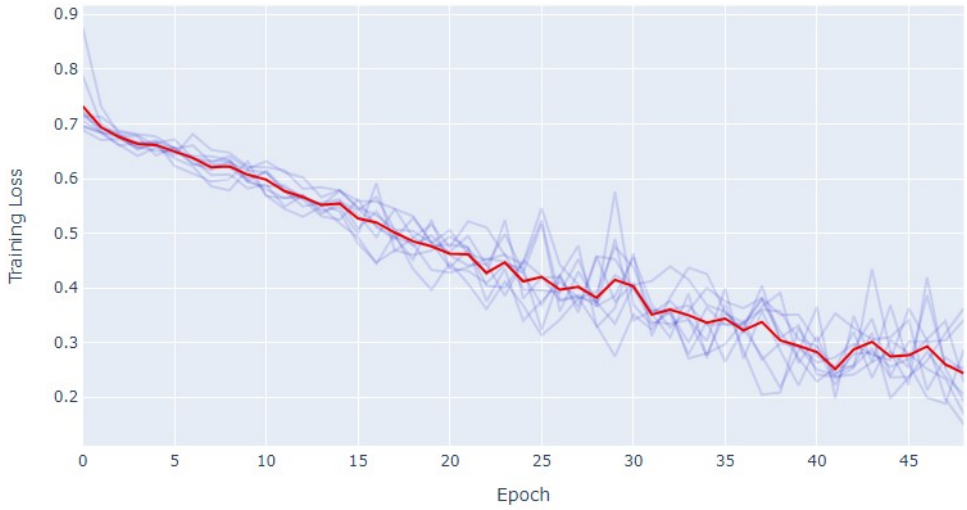

Figure 5.8: Training loss curve for overfitting CNN model

For the validation case, a well-known curve of overfitting can be observed in the validation loss vs. epoch graph (Figure 5.9).

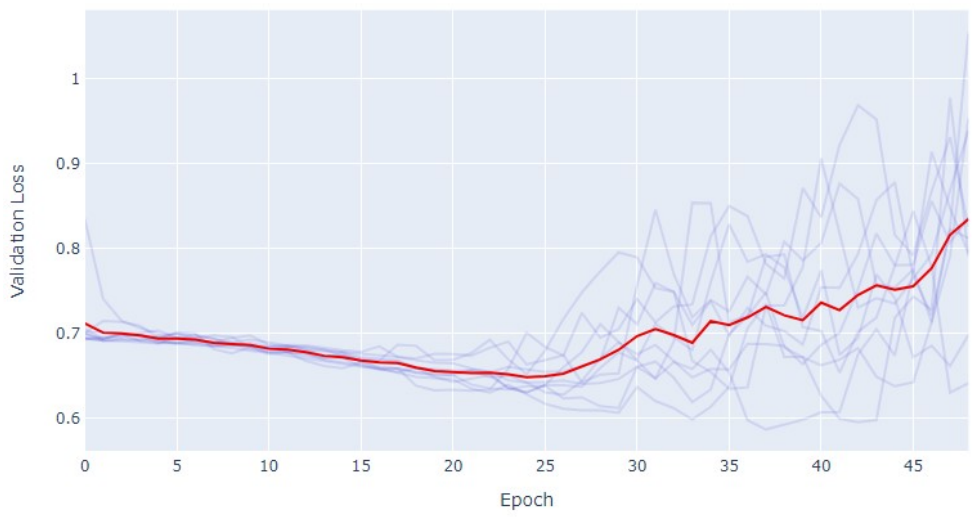

Figure 5.9: Validation loss curve for overfitting CNN model

As the validation loss starts increasing after a few number of epochs, validation accuracy also decreases as observed from Figure 5.10.

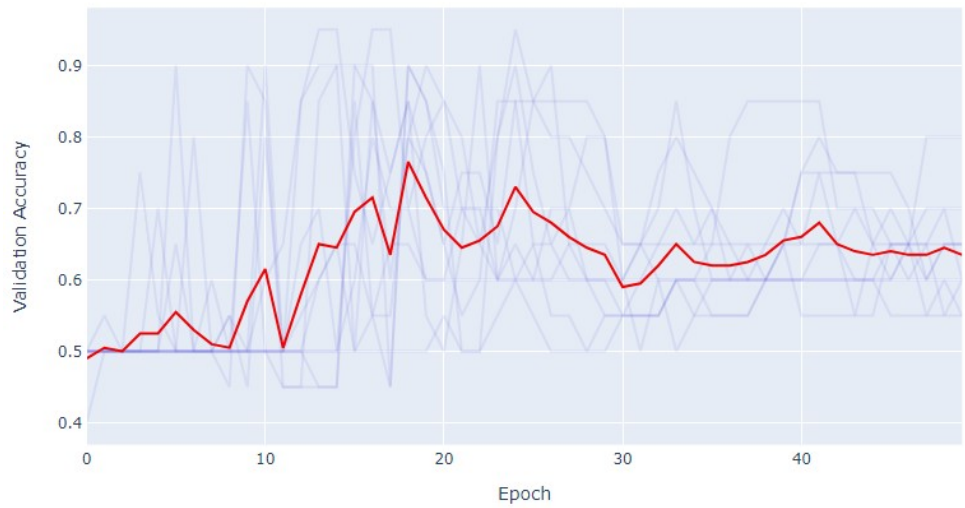

Figure 5.10: Validation accuracy curve for overfitting CNN model

After the network that overfits the input data is obtained, The search of the best hyperparameter configuration and stopping point for the training is started in order to get the best validation accuracy for fNIRS data to avoid overfitting.

Although the work proposed by Wang et al. [1] uses RELU for the activation func-

tion for the classification network, it is observed that the sigmoid activation function performs for the fNIRS data better with the proposed architecture. The reason behind this is due to the negative feature points that the fNIRS data have. These negative feature points are lost using the RELU activation function, and the accuracy decreases significantly. For the structure, some experimental changes are applied to the architecture suggested in [1]. 3 Convolutional Layers having 128 filters with a size of 16 are built, and they are convolved by a stride of 3. GAP layer is added to the end of the convolutional layers as proposed in order to summarize the channeled structure. The summary of the proposed model can be seen in Figure 5.11

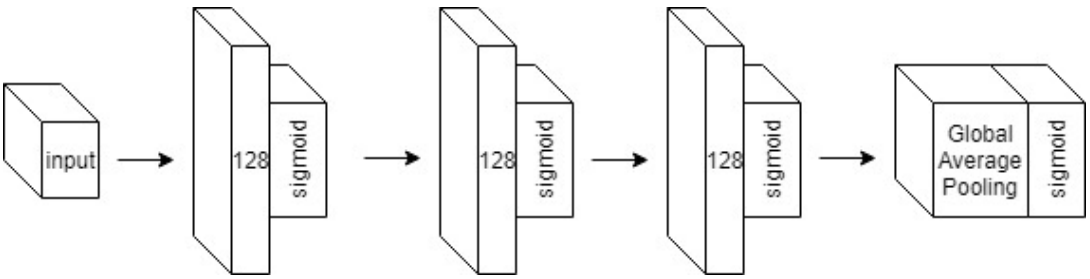

Figure 5.11: Summary of architecture for the proposed CNN model

The fast and efficient ADAM Optimizer is operated when estimation is being completed, and the learning rate was set to $3 * 10^{-5}$. Decay rates of the first and second moments are set to 0.8 and 0.9 for the optimizer. Mini batches of size five are used on training with batch normalization, and training is repeated 10 times with random initialization after discarding the trained model. Training and validation graphs are given in Figures 5.12, 5.13, 5.14, 5.15. In the plots, shaded curves represent a single repetition from 10 random initialization. The red curve is the average of the 10 repetitions. The training curves are obtained from a single fold of the five-fold cross-validation training iterations.

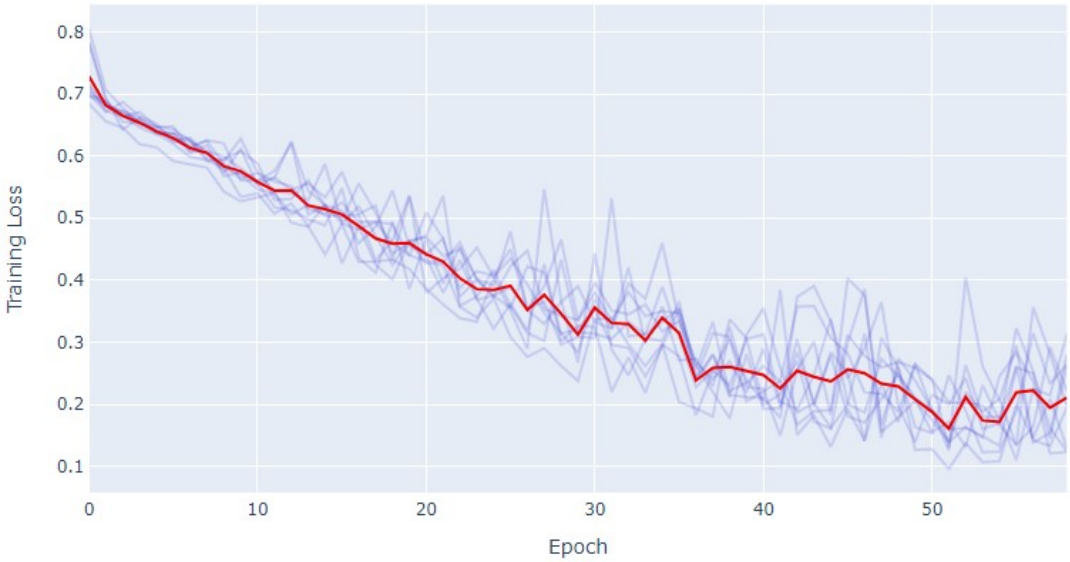

Figure 5.12: Training loss vs epoch for the proposed CNN model

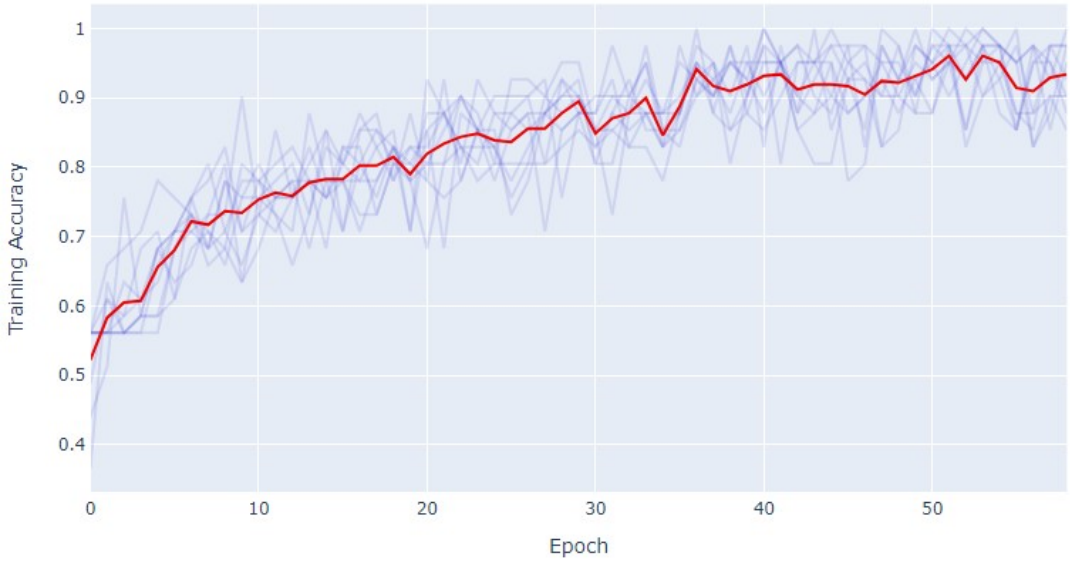

Figure 5.13: Training accuracy vs epoch for the proposed CNN model

As seen in Figure 5.12, the training loss decrease can be observed easily using the proposed model. Also, in Figure 5.13 is increasing to a point that satisfies the training.

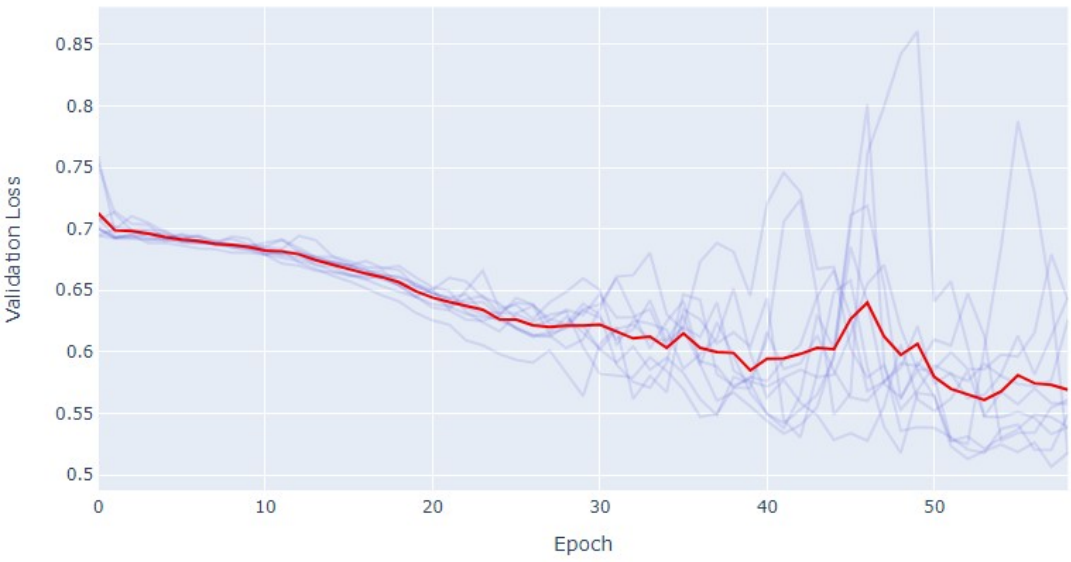

Figure 5.14: Validation loss vs epoch for the proposed CNN model

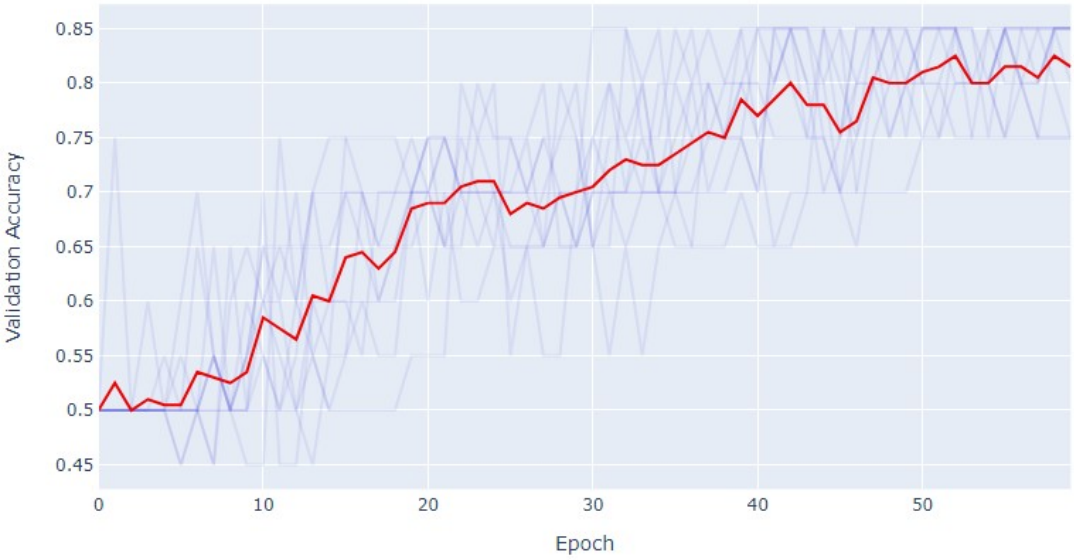

Figure 5.15: Validation accuracy vs epoch for the proposed CNN model

As it can be observed from the results, the network gives reasonably successful results on the validation data. The average accuracy on a sample validation set is 81.5%.

Although convolutional architectures were initially designed for visual classification tasks, one-dimensional modification of convolutional neural network architecture is

well trained on the fNIRS data, which is a multivariate time series. As is expected, the convolutional neural networks get ahead of the MLP on both validation and test sets.

After the hyperparameter tuning is completed, The networks are tested on 6 different test splits as it is done for MLP. The average result for different test splits is obtained as 73.25%. The results for each split can be observed from Table 5.2.

Table 5.2: Classification accuracy for each test split for the proposed CNN model

Split	1	2	3	4	5	6	μ	σ
Overall	71.8%	55%	74.7%	70.2%	85.95%	82%	73.27%	9.87
Bipolar	52%	50%	67%	60%	80%	80%	64.83%	12.06
Control	88.3%	75%	78%	77%	88.5%	84%	81.8%	5.41

A statistical representation of each test split is summarized in the box plot given in Figure 5.16. As observed from the plot, there is a distinct difference between the accuracy of bipolar and control subjects.

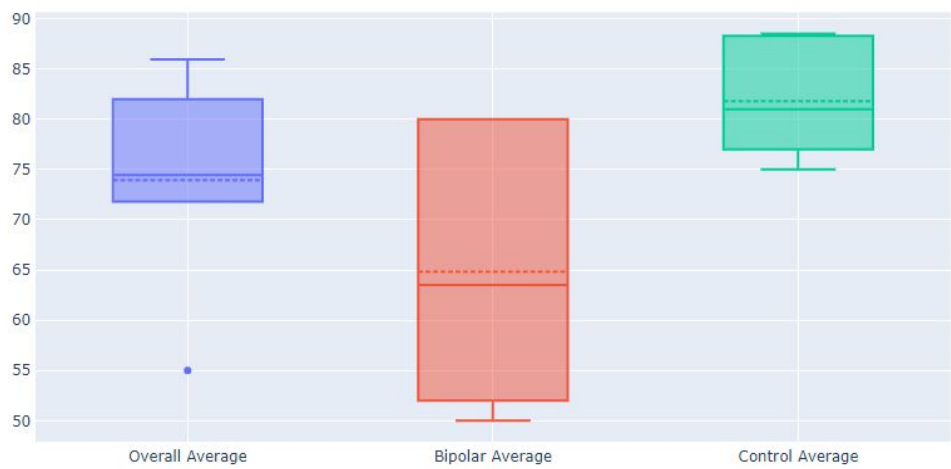

Figure 5.16: Box plot summary of the test set results for the proposed CNN model

5.1.3 Residual Neural Networks

After the CNN implementation, it is exposed that successful results can be attained using convolutional architectures on fNIRS data. Yet, as the network architecture gets deeper in layers, the accuracy decreases accordingly. Hence, a residual connection-based architecture inspired by time Residual Neural Network (ResNet) is selected to be the next model in order to assess deeper models without losing training capability. As seen in the implementation of CNN, the best performing model has three layers. After a while of getting more in-depth, CNN has lost its capacity to generalize the training set. The channeled temporal structure of fNIRS again allows us to use one-dimensional convolutional layers for the implementation. However, to populate more layers, shortcut layers are introduced to the network between some layers, which are chosen according to different trial and error cycles. For time ResNet structure best performing model consists of 6 layers, and shortcut connections are placed between Input-3 and 3-6 layers. It is also observed that shrinking kernel sizes (16, 8, 4) between residual connections work better for the fNIRS data. The summary of the ResNet architecture can be seen in Figure 5.17.

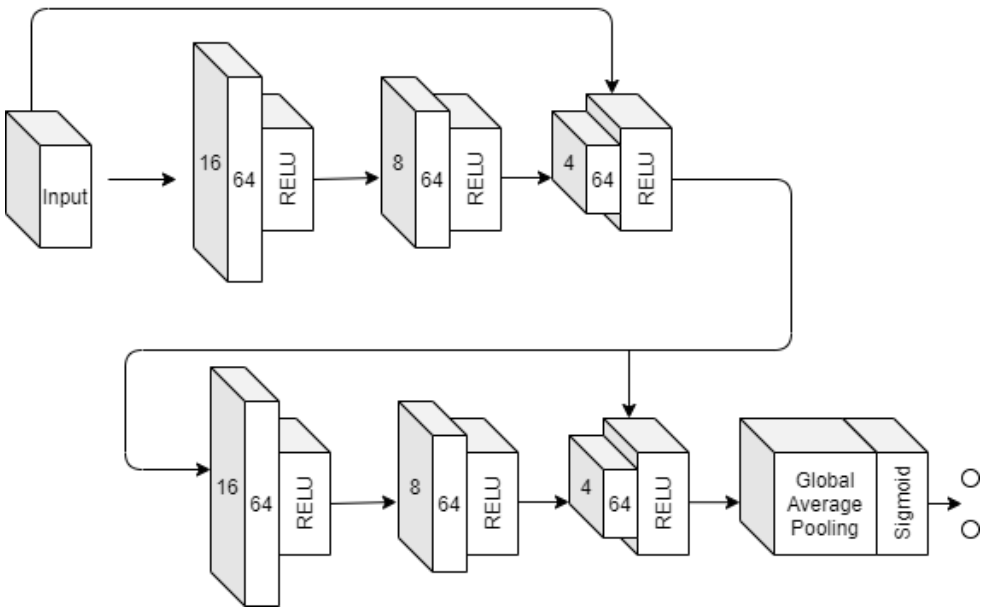

Figure 5.17: Summary of the proposed ResNet architecture

With the help of residual connections, the best performing model depth is increased to

6 layers, which were 3 for the CNN. This concludes the initial goal of using residual connections. The Loss and Validation graphs of a sample training iteration for the proposed model can be observed from the according Figures (5.18, 5.19, 5.20, 5.21).

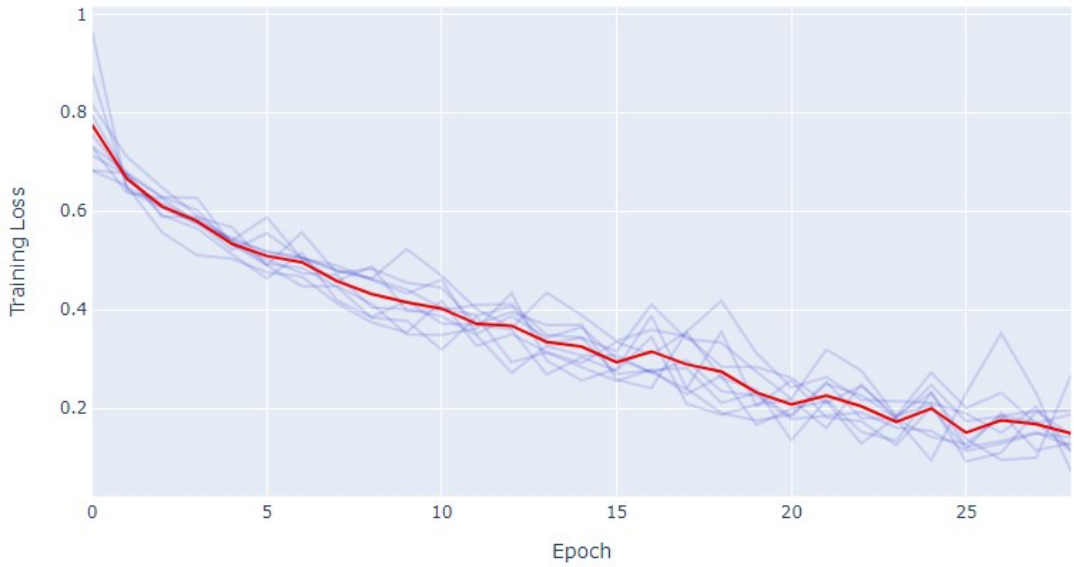

Figure 5.18: Training loss vs epoch for the proposed ResNet model

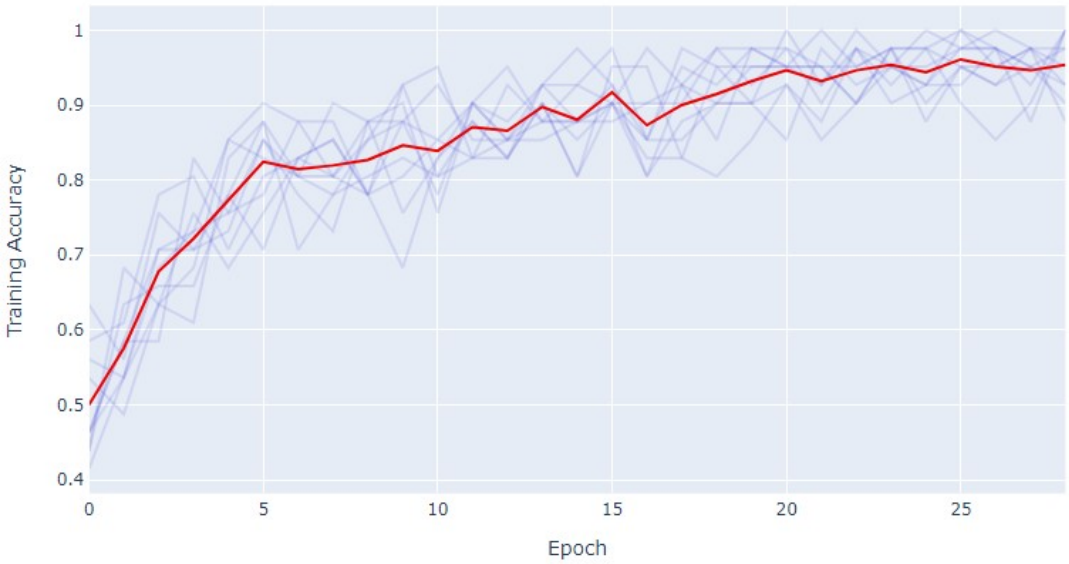

Figure 5.19: Training accuracy vs epoch for the proposed ResNet model

Training curves are observed without a problem meaning that the ResNet successfully generalizes the fNIRS data with the proposed architecture.

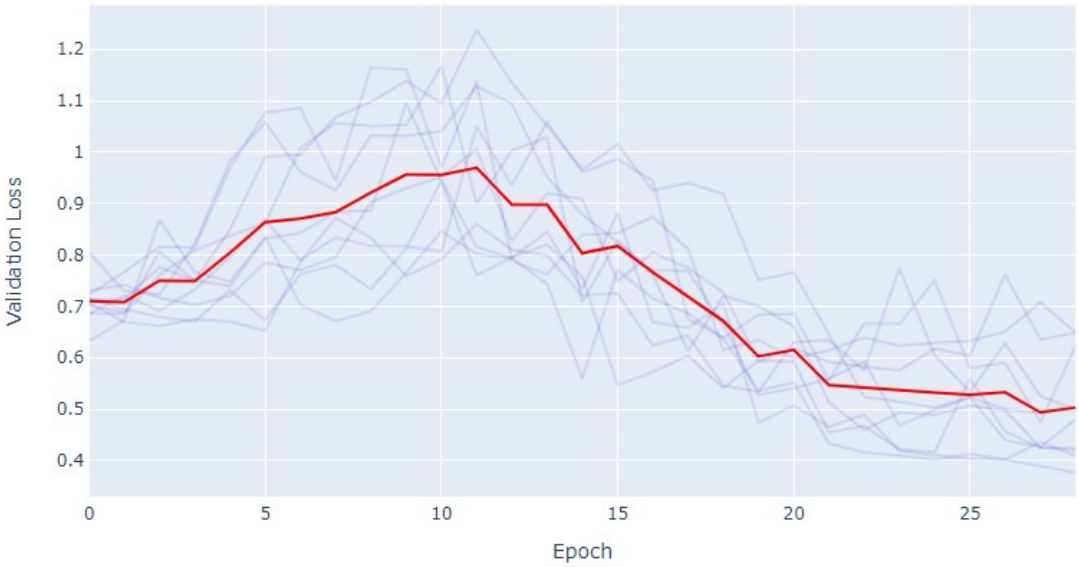

Figure 5.20: Validation loss vs epoch for the proposed ResNet model

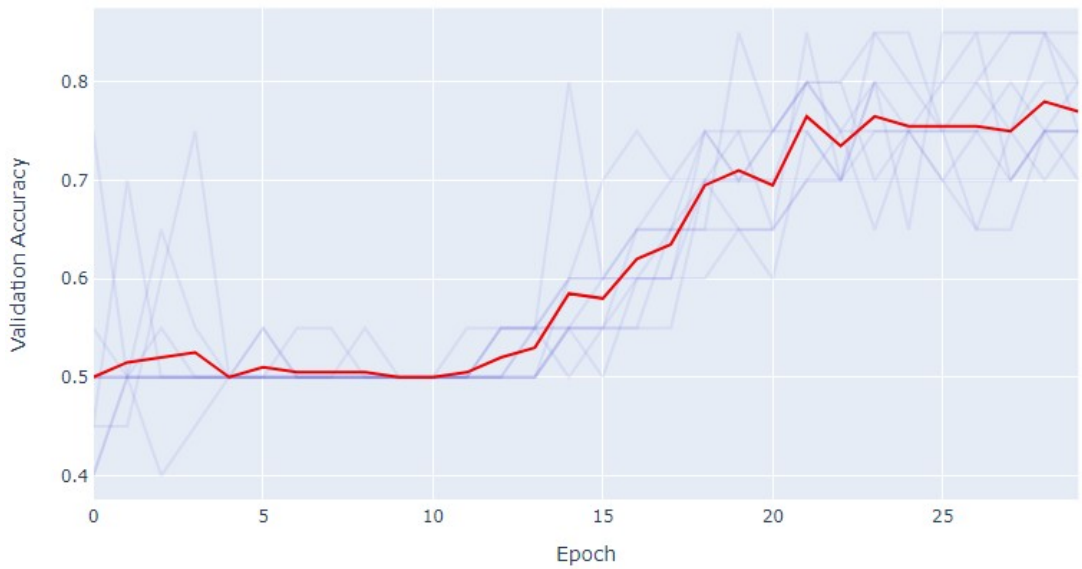

Figure 5.21: Validation accuracy vs epoch for the proposed ResNet model

Although the validation loss increases in the first part of the epoch cycles, after a while, the loss successfully converges to the global minima in terms of the validation loss and accuracy.

As it is applied for the CNN and MLP, the proposed residual connection-based archi-

tecture is also tested on all 6 test splits. As explained in the chapter 4, there exist 61 subjects, and all of these subjects are tested for each networks. Regarding this purpose, the subjects are divided into 6 different test folds. For each test fold, remaining 5 group are applied to 5 fold cross validation method. After each fold is tested, the model is discarded for the next test fold to be tested. The mean accuracy of 75.32% is obtained after the testing procedure. The result of each test split is exhibited in Table 5.3.

Table 5.3: Classification accuracy for each test split for the proposed ResNet model

Split	1	2	3	4	5	6	μ	σ
Overall	68.18%	60%	79.6%	80%	79.85%	85%	75.44%	8.56
Bipolar	60%	50%	67%	87.5%	83%	100%	74.58%	17.12
Control	75%	100%	85%	75%	78.5%	70%	80.58%	9.79

The results are also summarized in a box plot in Figure 5.22. The dashed lines in the plots represent the means for each class.

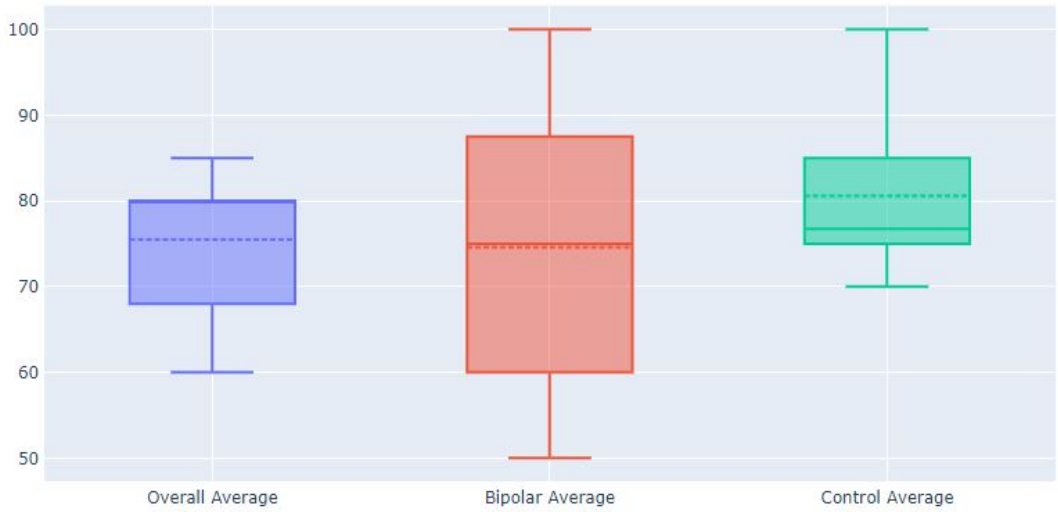

Figure 5.22: Box plot summary of the test set results for the proposed ResNet model

It can be concluded that the mean accuracy of 75.32% on the test sets is highly above

the chance level. Although the variance of bipolar subjects is still larger than the control group, the proposed residual connection-based architecture performs the best among all models. The mean classification accuracy and the distributions of all test splits demonstrate the fact that residual connection-based architectures get ahead of the pure convolutional layer based architectures.

5.1.4 Encoder Networks

Encoder networks are mainly used to cope with the variable sample length and the input data's multidimensionality. Although the variable length is not the case for this thesis's data, the effect of the attention mechanism is observed to get rid of the high dimensionality instead of using GAP layers. Moreover, the effect of the parametric RELU activation function is also observed. In the previous CNN-based architectures, the sigmoid function is chosen to be the activation function due to the negative feature points that occur in intermediate convolutional layers' activations. The parametric RELU function solves the mentioned problem by introducing a slope at the negative part of the RELU.

The best performing encoder model has three convolution layers with 64 filters in each layer, followed by parametric RELU activation functions. Dropout, with a rate of 0.05 is used to prevent overfitting. After the third convolutional layer, the output filters are divided into two parts to applying the attention mechanism. The training and validation curves of a sample validation fold can be observed from Figures 5.23 , 5.24, 5.25, 5.26.

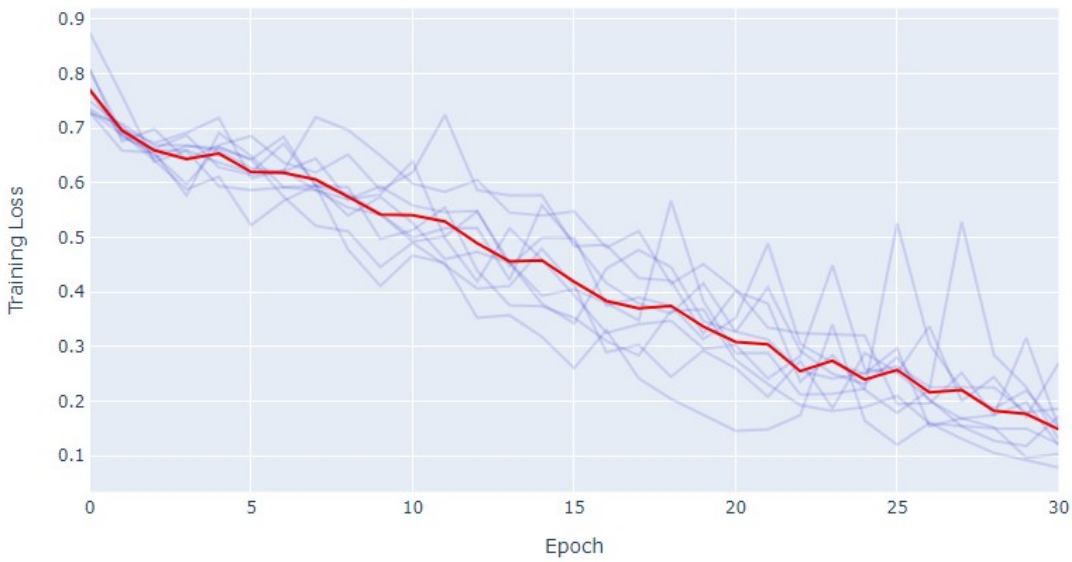

Figure 5.23: Training loss vs epoch for the proposed Encoder model

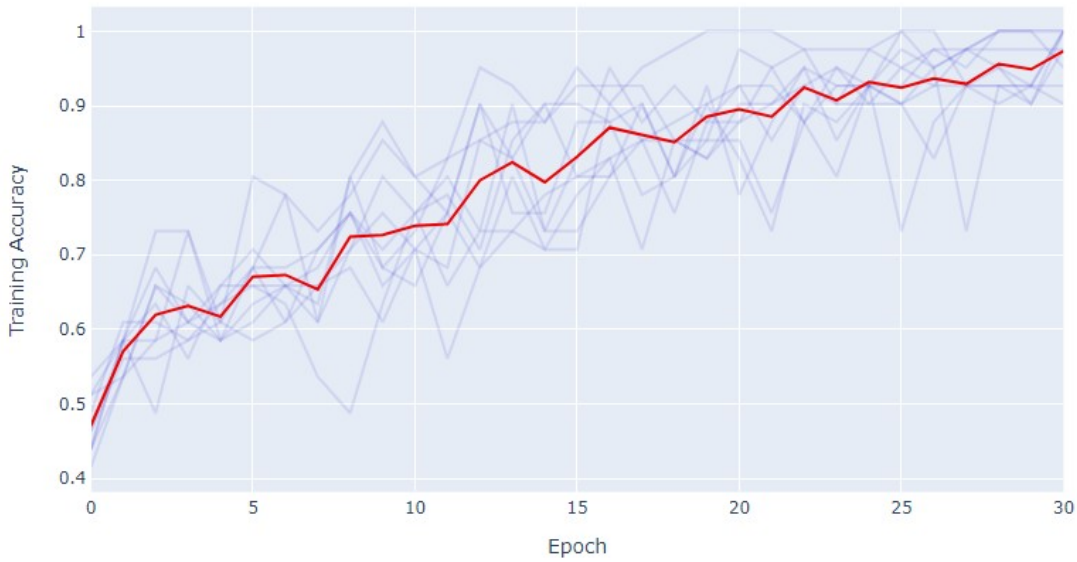

Figure 5.24: Training accuracy vs epoch for the proposed Encoder model

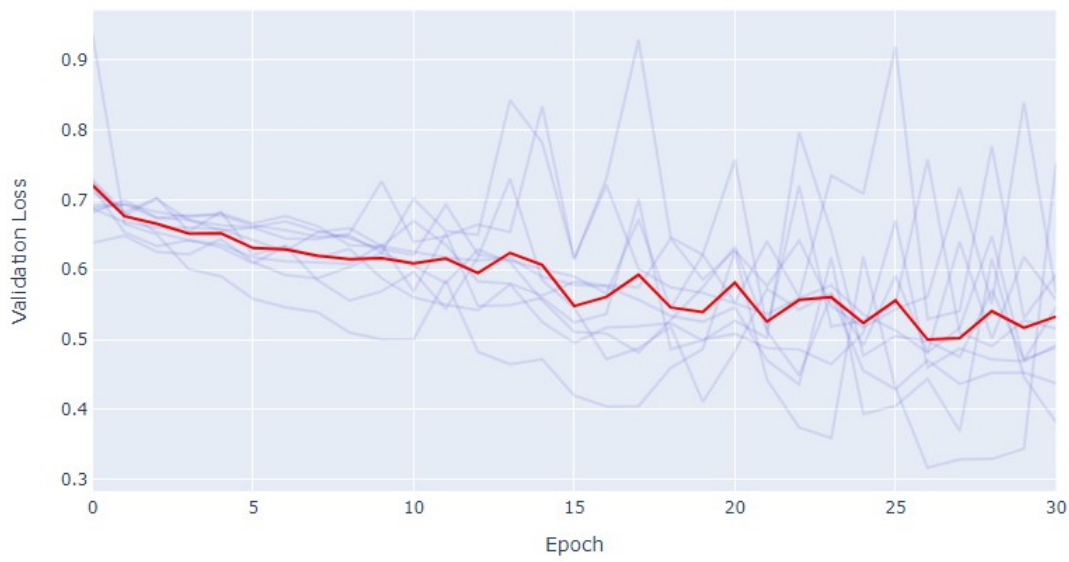

Figure 5.25: Validation loss vs epoch for the proposed Encoder model

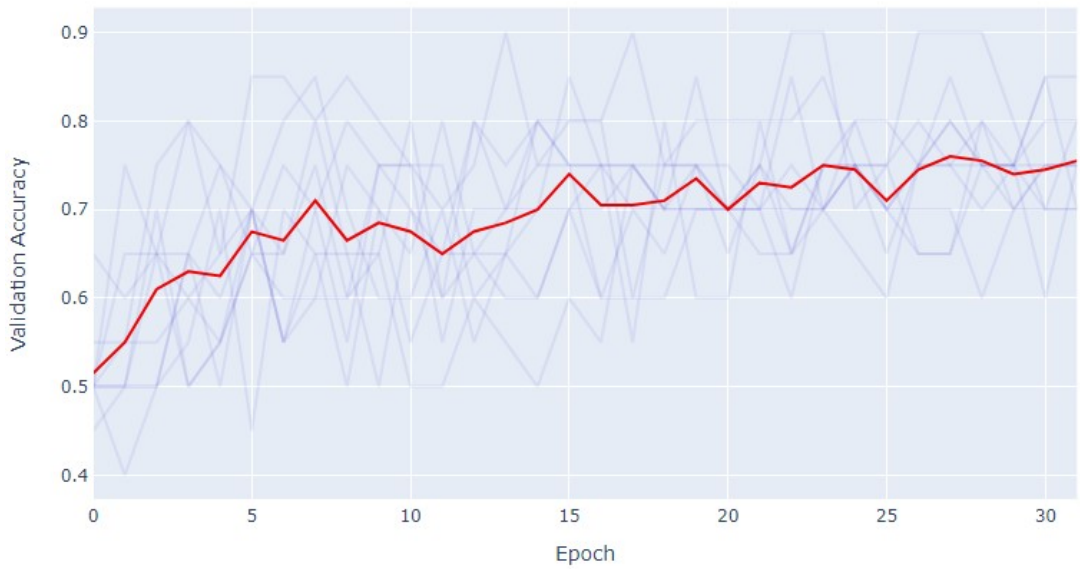

Figure 5.26: Validation accuracy vs epoch for the proposed Encoder model

For the best performing encoder model considering the validation accuracy and loss plots, it can be said that using an attention mechanism to summarize temporal features on a time series is comparably successful as using the global average pooling method. Moreover, it can be concluded that Encoder networks can be used to classify fNIRS data.

When the proposed Encoder network is tested on the test sets as the previous models, it is seen that proposed networks based on attention mechanism are not as successful as proposed CNN and ResNet based methods. However, the models still classify the fNIRS data highly above the chance level. The model is tested with the same procedure as CNN and ResNet using the 6 different test splits and performs a mean accuracy of 71.22%. The results are exhibited in Table 5.4.

Table 5.4: Classification accuracy for each test split for the proposed Encoder model

Split	1	2	3	4	5	6	μ	σ
Overall	72.54%	51.6%	74.92%	74.2%	74.94%	74%	70.28%	8.9
Bipolar	60%	52%	66.6%	40%	83.3%	82%	63.98%	15.49
Control	83%	50%	78.5%	97%	78.5%	66%	75.5%	14.59

Although the network can still be considered successful, the CNN and ResNet based architectures outperform the Encoders. The test set results are statistically summarized in Figure 5.27

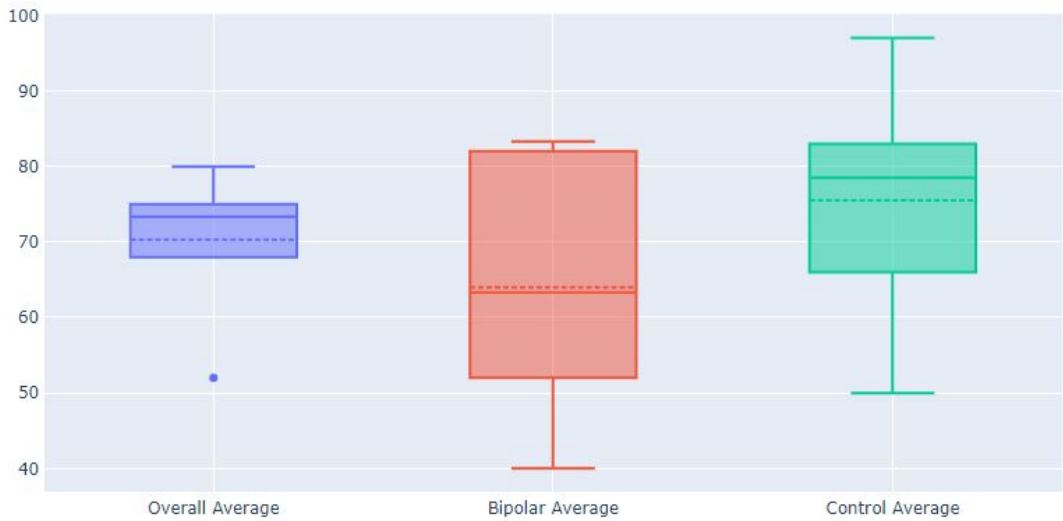

Figure 5.27: Box plot summary of the test set results for the proposed Encoder model

5.1.5 Test Set Comparison

Although the number of usable subjects in the dataset is not high enough for deep learning methods, it is observed that quite successful results are obtained using the mentioned neural network architectures. In order to compare the architectures in detail, as it is mentioned in earlier chapters, 6 different test split is arranged for test purposes. For each test split, 5 fold cross-validation is applied, The best resulting fold is tested on the test set, and the results are exhibited in earlier chapters. Although performances of all proposed architectures are high above the chance level, the performance of the most fundamental artificial intelligence architecture MLP is unsuccessful relative to the other models. The accuracy, the sensitivity (true positives for the BD group) and the specificity (true negatives for the control group) results are gathered together in Table 5.5.

Table 5.5: Classification accuracy for each model on the test set

TEST METHOD	MLP	CNN	ResNet	Encoder
Accuracy	64.45%	73.25%	75.32%	71.22%
Sensitivity	60.68%	62.17%	71.43%	61.99%
Specificity	67.68%	82.64%	78.62%	79.06%

The overall results imply that for the complete dataset, the most successful neural networks are ResNets, which are followed by CNN's. From the results, it can be said that the Encoders and attention mechanism is not as successful as the Global Average Pooling layers for the average test set splits. However, it should be noted that in the future, as the number of the samples on the dataset increases, the comparison between the models will become clearer.

5.1.6 Discussions

The baseline of the MLP, CNN and ResNet structures are firstly proposed by Wang et al. [1]. In this study, they concluded that the CNN architecture achieves the premium

performance among the models. They also stated that ResNet structure also performs comparable performance in the experiments. The MLP structure differs from the CNN and ResNet since the network is not based on convolutional layers. Therefore, spatial information gets lost. This means the network executes the time series elements as independent elements. This causes poor classification performance among the other models. When the fNIRS input results are checked, MLP is the worst performing neural network with an accuracy of 64.45%. This result is consistent with the baseline research. The ResNet and CNN structures' accuracy are 75.32% and 73.25%, respectively, which makes them the best among all models. The last neural network suggested in this thesis' scope is the Encoder network. Similar to both CNN and ResNet, the Encoder network also uses convolutional layers for the feature extraction. The main difference between these networks is that the Encoder utilizes a mechanism called the attention, while CNN and ResNet use GAP layers to summarize the temporal channels.

In the research suggested by Fawaz et al. [21], the mentioned networks are compared with several datasets. When evaluated over 85 univariate and 12 multivariate time series datasets, ResNet outperforms other networks in the majority of these datasets. For the fNIRS data, ResNet is also the best performing classifier in terms of accuracy, followed by CNN. Furthermore, the same work states that ResNet successfully highlights discriminative regions by using Class Activation Maps (CAM). ResNet's ability originated from the residual connections between layers allowing the model to learn to skip unnecessary convolutions. They also suggest that it occurs, so the ResNet outperforms CNN. Although the both ResNet and CNN perform successfully for fNIRS data, ResNet performs slightly better, therefore, our results are consistent for both works. It might be deduced that the GAP layer outperforms the attention mechanism for fNIRS data. This finding also supports the evidences from the research of Fawaz et al. However, results having significant accuracies can be obtained with the use of the attention mechanism. Hence, THE Encoder network can be suggested as an alternative for the ResNet and CNN while using the fNIRS data. Since the study mentioned had been a big inspiration for this thesis, acquiring consistent results on our experimental fNIRS dataset is a big achievement that should be considered.

It should be stated that random initializations affect the performance significantly

for various datasets [21]. Although the experiments for each neural network were repeated 10 times with random initialization of the networks, the results can slightly change with different experiments in our study.

It is observed for all architectures that the classification of bipolar subjects is less successful than that of control subjects. However, our expertise on the fNIRS data is insufficient to specify the reason behind this abnormality in the sensitivity (true positives for BD) of the classification methods. Some studies [34, 35] show that there exist some inconsistencies in the reported changes in the frontal lobe function during an activation task for depressed BD patients. The original data [29] was taken from patients in the same stage of the disease. Hence, it can be concluded that the illness state does not cause the lower sensitivity of the classification. However, the reason might be the severity of the disorder on each subject. Furthermore, when the activations of the last layers of neural networks are investigated in the study of Babacan [36], it can be seen that the activations of the BD subjects are distributed over a wider range of values which can also be the reason behind the less classification accuracy compared to the control group.

Overall, the results support the view that the ResNet is determined as the most accurate neural network with a classification accuracy of 75.32%, the sensitivity of 71.43%, and the specificity of 78.62%. As it is reviewed in the literature, in the work proposed by Frangou et al.[7], the BD is classified using the pattern classification with the help of task-based functional magnetic resonance imaging (fMRI). In the work, they correctly classified 30 bipolar subjects with 30 unrelated healthy individuals by the accuracy of 83.5%, the sensitivity of 84.6%, and the specificity of 92.3%. When both results are compared, using the fNIRS data with deep neural networks for the classification of BD is slightly outperformed by the use of fMRI data. However, It is vital to bear in mind that the individuals used for both experiments are entirely different.

To the best of our knowledge, the only study that used fNIRS to classify BD is the work proposed by our research group [2]. The study categorized the BD with an accuracy of 69%. In the work, our research group operated in only the resting period of the VF task. The network suggested in that work used the fully connected layers

to summarize the temporal information at the end of the convolutional layers. In this thesis's scope, these layers are replaced with GAP layers for the CNN and ResNet based architectures and an attention mechanism for the Encoder based architecture. Another difference of the proposed work is for the evaluation of the neural network on the fNIRS data, the leave one out method (LOO) is chosen with the intent of using more data in the training set. However, it is observed that using LOO method accompanies the possibility of a training leakage. Moreover, by the end of this thesis, the classification performance of the previous work is outperformed by several architectures.

In conclusion, a considerable amount of literature has been published on time series classification with deep learning using several methods on various datasets. The findings of this research on fNIRS data seem to be in agreement with the previous literature on time series classification with deep learning. Although this research seems to be related with only fNIRS data classification, these networks can be modified and used for every kind of time series data that is going to be researched.

CHAPTER 6

CONCLUSIONS

6.1 Summary and Conclusion

In this study, the mentioned deep neural network structures that are specialized for time series classification are trained with the fNIRS measurements taken from verbal fluency test, which is explained in chapter 4.

The first goal that was achieved is obtaining an acceptable classification accuracy using the data available. In the first part of the study, in order to use fNIRS measurements as an input to our neural networks, the preparation and the identification of the not usable data is completed. After the extraction of the not usable data, the neural networks that are specialized for time series classification are designed and modified to use fNIRS measurements. For this purpose, 4 different neural networks (MLP, CNN, ResNet, Encoder) are chosen. Since the neural networks designed for time series classification diverges from the ones for visual classification tasks as the number of layers, the networks are designed, modified, and implemented accordingly. This means the resulting neural networks have fewer intermediate layers than the ones for visual classification tasks.

Since the test set sampling is an issue that should be worked intensively, for a fair comparison, the complete dataset is separated into 6 different test splits, and 6 different neural networks are trained and validated for each of the splits. Validation for each split is also done by 5 fold cross validation with the remaining 5 splits of the data. After the 5 fold cross-validation, the best fold is tested on the test split. The procedure is repeated for each test split by discarding the trained networks for fair comparison among the models. It was seen that ResNet is the best performing model among the

tested architectures with an accuracy of 75.32% . However, the other state of the art methods (CNN and Encoder) results were also very successful with the result percentages 73.25% and 71.22%. The multilayer perceptron method that is applied was not successful enough and fell behind among other models. The reason behind this is to apply the multi-channel input to Multilayer Perceptrons. The data is flattened over 24 channels. This is not the best solution for designing a neural network for a multivariate time series classification.

In conclusion, the classification success implies that using deep neural networks that are specialized for multivariate time series classification is critically useful for the classification of fNIRS data for BD classification, and this will pave the way for new works related to the work done in this study.

6.2 Related and Future Works

In the scope of this thesis, the classification of BD using fNIRS measurements has been implemented. However, for verification and statistical validation of the neural network and regarding training outputs is needed to be done. Another member of our research group focused on this issue and evaluated the results in the thesis to be published by him [36]. After the implementation of the neural network activation visualization by using state of the art neural network visualization methods, the outcomes are investigated with the help of statistical tests, namely chi-square test, and t-test. The findings infer that distributions of the activation maps of both bipolar and healthy subjects are homogeneous in themselves, and the results hang together; at the same time, they are distinguishable from each other. Hence, both works coincide with each other and cross-check themselves.

Although the researches mentioned concludes a successful classification for bipolar disease using fNIRS measurements. Still, future work with additional data is needed since the data collection is not sufficient enough. As more the data collection gets, the results will get healthier for both types of research.

The channeled structure, therefore, the connections and priorities between these channels are a significant issue. Another future work for our study group is to design some

networks that can reveal the importance of distinct channels and tracing the neural network successes in the light of new information that is going to be obtained. This information will lead to many studies with our research group.

REFERENCES

[1] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," 2016.

[2] H. B. Evgin, O. Babacan, I. Ulusoy, Y. Hoşgören, A. Kuşman, D. Sayar, B. Baskak, and H. D. Özgüven, "Classification of fnirs data using deep learning for bipolar disorder detection," in *2019 27th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, 2019.

[3] M. Azechi, M. Iwase, K. Ikezawa, H. Takahashi, R. Kurimoto, T. Nakahachi, R. Ishii, M. Fukumoto, K. Ohi, Y. Yasuda, H. Kazui, R. Hashimoto, and M. Takeda, "Discriminant analysis in schizophrenia and healthy subjects using prefrontal activation during frontal lobe tasks: A near-infrared spectroscopy," *Schizophrenia research*, vol. 117, pp. 52–60, 11 2009.

[4] Y. Monden, I. Dan, M. Nagashima, H. Dan, M. Uga, T. Ikeda, D. Tsuzuki, Y. Kyutoku, Y. Gunji, D. Hirano, T. Taniguchi, H. Shimoizumi, E. Watanabe, and T. Yamagata, "Individual classification of adhd children by right prefrontal hemodynamic responses during a go/no-go task as assessed by fnirs," *NeuroImage: Clinical*, vol. 9, pp. 1 – 12, 2015.

[5] Y. Akira, M. Omori, A. Fukuda, J. Takahashi, Y. Yasumura, E. Nakagawa, T. Koike, Y. Yamashita, T. Miyajima, T. Koeda, M. Aihara, H. Tachimori, and M. Inagaki, "Applied machine learning method to predict children with adhd using prefrontal cortex activity: A multicenter study in japan," *Journal of Attention Disorders*, p. 108705471774063, 11 2017.

[6] N. Jie, M. Zhu, X. Ma, E. A. Osuch, M. Wammes, J. Théberge, H. Li, Y. Zhang, T. Jiang, J. Sui, and V. D. Calhoun, "Discriminating bipolar disorder from major depression based on svm-foba: Efficient feature selection with multimodal brain imaging data," *IEEE Transactions on Autonomous Mental Development*, vol. 7, no. 4, pp. 320–331, 2015.

[7] S. Frangou, D. Dima, and J. Jogia, "Towards person-centered neuroimaging markers for resilience and vulnerability in bipolar disorder," *NeuroImage*, vol. 145, 09 2016.

[8] D. Librenza-Garcia, B. J. Kotzian, J. Yang, B. Mwangi, B. Cao, L. N. Pereira Lima, M. B. Bermudez, M. V. Boeira, F. Kapczinski, and I. C. Passos, "The impact of machine learning techniques in the study of bipolar disorder: A systematic review," *Neuroscience Biobehavioral Reviews*, vol. 80, pp. 538 – 554, 2017.

[9] F. Jöbsis, "Noninvasive, infrared monitoring of cerebral and myocardial oxygen sufficiency and circulatory parameters.," *Science*, vol. 198 4323, pp. 1264–7, 1977.

[10] M. Ferrari, L. Mottola, and V. Quaresima, "Principles, techniques, and limitations of near infrared spectroscopy," *Canadian journal of applied physiology = Revue canadienne de physiologie appliquée*, vol. 29, pp. 463–487, 08 2004.

[11] J. Leon-Carrion and U. Leon-Dominguez, *Functional Near-Infrared Spectroscopy (fNIRS): Principles and Neuroscientific Applications*, pp. 47–74. 01 2012.

[12] D. A. Boas, A. M. Dale, and M. A. Franceschini, "Diffuse optical imaging of brain activation: approaches to optimizing image sensitivity, resolution, and accuracy," *NeuroImage*, vol. 23, pp. S275 – S288, 2004. Mathematics in Brain Imaging.

[13] H. Kim, K. Seo, H. Jeon, U. Lee, and H. Lee, "Application of functional near-infrared spectroscopy to the study of brain function in humans and animal models," *Molecules and cells*, vol. 40, 08 2017.

[14] D. J. Miklowitz and S. L. Johnson, "The psychopathology and treatment of bipolar disorder," *Annual Review of Clinical Psychology*, vol. 2, no. 1, pp. 199–235, 2006. PMID: 17716069.

[15] V. Nair and G. Hinton, "Rectified linear units improve restricted boltzmann machines vinod nair," vol. 27, pp. 807–814, 06 2010.

[16] D. E. Rumelhart and J. L. McClelland, *Learning Internal Representations by Error Propagation*, pp. 318–362. 1987.

[17] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.

[18] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin, "The elements of statistical learning: Data mining, inference, and prediction," *Math. Intell.*, vol. 27, pp. 83–85, 11 2004.

[19] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, p. 1929–1958, Jan. 2014.

[20] M. Lin, Q. Chen, and S. Yan, "Network in network," 2014.

[21] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data Mining and Knowledge Discovery*, vol. 33, p. 917–963, Mar 2019.

[22] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," *Neural Information Processing Systems*, vol. 25, 01 2012.

[23] H. Lee, R. Grosse, R. Ranganath, and A. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," p. 77, 01 2009.

[24] J. C. B. Gamboa, "Deep learning for time-series analysis," 2017.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[26] J. Serrà, S. Pascual, and A. Karatzoglou, "Towards a universal neural network encoder for time series," 2018.

[27] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," 2017.

[28] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2016.

[29] Y. H. ALICI, *Bipolar Bozukluk Hastalarında Sözel Akıcılık Ve Düşünce Akıcılığı Performansları Sırasında Prefrontal Korteks Aktivitesinin Psikososyal Uyum, İşlevsellik, Yaşam Kalitesi Ve İç Görü İle İlişkisi*. PhD thesis, Ankara Üniversitesi Tıp Fakültesi, 7 2015.

[30] M. Lezak, "Neuropsychological assessment, 3rd ed.," 1995.

[31] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," 2016.

[32] F. Chollet *et al.*, "keras," 2015.

[33] D. Stathakis, "How many hidden layers and nodes?," *International Journal of Remote Sensing*, vol. 30, no. 8, pp. 2133–2147, 2009.

[34] S. M. Strakowski, M. P. DelBello, C. Adler, K. M. Cecil, and K. W. Sax, "Neuroimaging in bipolar disorder," *Bipolar disorders*, vol. 2, no. 3, pp. 148–164, 2000.

[35] M. Kameyama, M. Fukuda, Y. Yamagishi, T. Sato, T. Uehara, M. Ito, T. Suto, and M. Mikuni, "Frontal lobe function in bipolar disorder: A multichannel near-infrared spectroscopy study," *NeuroImage*, vol. 29, pp. 172–84, 02 2006.

[36] O. Babacan, "Visualization of deep networks trained for bipolar disorder classification by using fnirs measurements," Master's thesis, The Graduate School of Natural and Applied Sciences of Middle East Technical University, 2 2021.