Auditory and Visual Rhythm Omission EEG Classification using Hybrid TLBO and Differential Evolution based Channel Selection

Submitted in partial fulfilment of the requirements of the degree of

Bachelor of Technology (B.Tech) by

Aniket Dasurkar (197109)
Swarna Sweth (197181)
Jatavath Surya Prakash (197237)

Supervisors:

Dr. Ramalingaswamy Cheruku (Supervisor)

Department of Computer Science and Engineering

NIT Warangal

India

Certificate

This is to certify that the Dissertation work entitled **"EEG Signal Classification using Deep Learning"** is a bonafide record of work carried out by **"Aniket Dasurkar (197109), Swarna Sweth (197181), Jatavath Surya Prakash (197237)"**, submitted to Dr. Ramalingaswamy Cheruku of "Department of Computer Science and Engineering", in partial fulfilment of the requirements for the award of the degree of B.Tech at "National Institute of Technology, Warangal" during the 2022-2023.

Dr. Ramalingaswamy Cheruku
Supervisor
"Department of Computer Science and Engineering"

Abstract

Electroencephalograms can be used to assess conditions relating to the brain (EEG). The process of manually analyzing EEG data is recognized to have a comparatively low level of accuracy and requires significant expertise from qualified professionals. The current manual approach for analyzing EEG data is known to be both resource-intensive and time-consuming due to the large amount of data involved and the constant influx of new data. Furthermore, this approach requires the expertise of trained professionals and can be expensive. However, automated EEG data processing has the potential to significantly enhance cognitive analysis for various applications by reducing human error and speeding up diagnosis. EEG is also a powerful tool for studying cognitive functions and helps to better understand the neural mechanisms underlying cognitive processes. This research focuses on studying how the brain behaves in responses to omissions in rhythmic stimuli using event related potentials or ERPs. This can also help ongoing research on developing new paradigms and analysis techniques to better understand the neural mechanisms underlying these processes. The presented approach can be applied to several EEG datasets by employing **deep learning** techniques using **temporal convolution networks**. A variety of applications are explored here, including the use of EEG toward diagnosing cognitive disorders such as ADHD, epilepsy and schizophrenia. A non-healthcare application would be that of lie detection who's use case can be pivotal to criminal detection.

Keywords - Electroencephalograms (EEG), Teaching Learning Based Optimization (TLBO), Differential Evolution (DE), Temporal Convolutional Networks (TCN)

Contents

List of Figures

Chapter 1

Introduction

Electroencephalography (EEG) is a test to measure brain activity through the recording of spontaneous electrical activity across a patient's scalp and is a popular tool used for the diagnosis of various neurological conditions and disorders. In this procedure, a number of small electrodes attached with thin wires are affixed to the scalp to record the electrical charges generated by the brain cells' activity. EEG is a commonly employed diagnostic tool in medical practice due to its excellent temporal resolution, affordability, and non-invasive temperament.

EEG-based studies of cognitive functions typically involve presenting participants with various types of stimuli, such as images, sounds, or words, and analyzing the patterns of neural activity that are elicited in response.

Auditory omission stimuli can include the omission of a tone in a repetitive tone sequence, while visual omission stimuli can include the omission of a flashing light in a repetitive light sequence. The brain's response to omission stimuli is typically measured using event-related potentials (ERPs) and time-frequency analysis techniques. One of the most studied ERPs in response to omission stimuli is the mismatch negativity (MMN), which is a negative waveform that occurs approximately 100-250 milliseconds after the omission of an expected stimulus. The MMN is thought to reflect the brain's automatic detection of a deviant or unexpected stimulus, and is believed to be related to processes involved in sensory memory and attention. Large volumes of data are produced during this procedure, which require skilled investigators to manually interpret. Therefore, we introduce a channel selection algorithm which is a hybrid of two popularly known channel selection algorithms called Teaching Learning Based Optimization(TLBO) and Differential Evolution (DE) followed by deep learning methods for classification. EEG interpretation is a laborious process that can cause large delays during treatment due to the shortage of certified expert investigators and the large volume of data. By speeding up the reading process and hence decreasing workload, adding a certain amount of automation to the EEG interpretation duty could be helpful to neurologists.

Chapter 2

Literature Review

2.1 Electroencephalograms

The paper "An Efficient Concealed Information Test: EEG Feature Extraction and Ensemble Classification for Lie Identification" by Annushree Bablani et al.[5] presents an innovative approach to the use of EEG signals in identifying deception. However, this paper also serves as a useful introduction to the world of EEG signal processing, and it offers insights into the various techniques and methods used in EEG feature extraction and classification.

This provided a clear introduction to EEG signals and their potential applications. The study's methodology and results are significant contributions to the field of EEG signal processing and can serve as a foundation for future research in this area.

2.2 Channel Selection Algorithms

The field of optimization has seen significant advancements in recent years, with various algorithms proposed for solving optimization problems in different domains. Two such algorithms are Teaching-Learning-Based Optimization (TLBO) and Differential Evolution (DE), which have shown promising results in solving complex optimization problems.

R. Venkata Rao proposed TLBO in his paper "Teaching-Learning-Based Optimization Algorithm and Its Engineering Applications."[12] This algorithm is based on the concept of teaching and learning, where individuals in a population are considered as teachers and learners.

In the paper, "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces,"[13] Rainer Storn and Kenneth Price proposed the Differential Evolution (DE) algorithm. DE is a population-based algorithm that uses the concept of mutation and crossover to generate new individuals in the population.

Both TLBO and DE have been used in various optimization problems, and they have shown promising results, thus the combination of such algorithms has great potential.

Chapter 3

Background

In this section, we discuss how binary TLBO[3] is used for channel selection and the working principle of recurrent neural networks, gated recurrent units and inception modules which we have used in implementing Chrononet and ResNet models.

3.1 TLBO

According to Rao[12], TLBO is a meta-heuristic optimization algorithm that is based on the simulation of the teaching and learning methodology. It mimics the teaching and learning processes that occur in a classroom to improve the performance of the students.

In binary TLBO[3], a population of candidate feature subsets is generated, and the best performing subsets are selected based on their fitness scores. The algorithm then performs a teaching process where the best performing subset shares its features with other subsets in the population. This is followed by a learning process where each subset adjusts its features based on the shared information from the teaching phase.

The algorithm iterates through these two phases until a stopping criterion is met, and the best performing subset is selected as the optimal feature subset. Overall, binary TLBO can effectively select a subset of features that maximizes the accuracy of the classification model while minimizing the number of channels used.

3.2 Differential Evolution

Differential Evolution (DE) [13] is a heuristic optimization algorithm that can be used for channel selection. The basic idea of DE is to generate new candidate solutions by combining the information from existing solutions in the population. The DE algorithm employs a technique wherein a new potential solution is created by taking a weighted difference between two randomly chosen solutions from a set of three or more so-

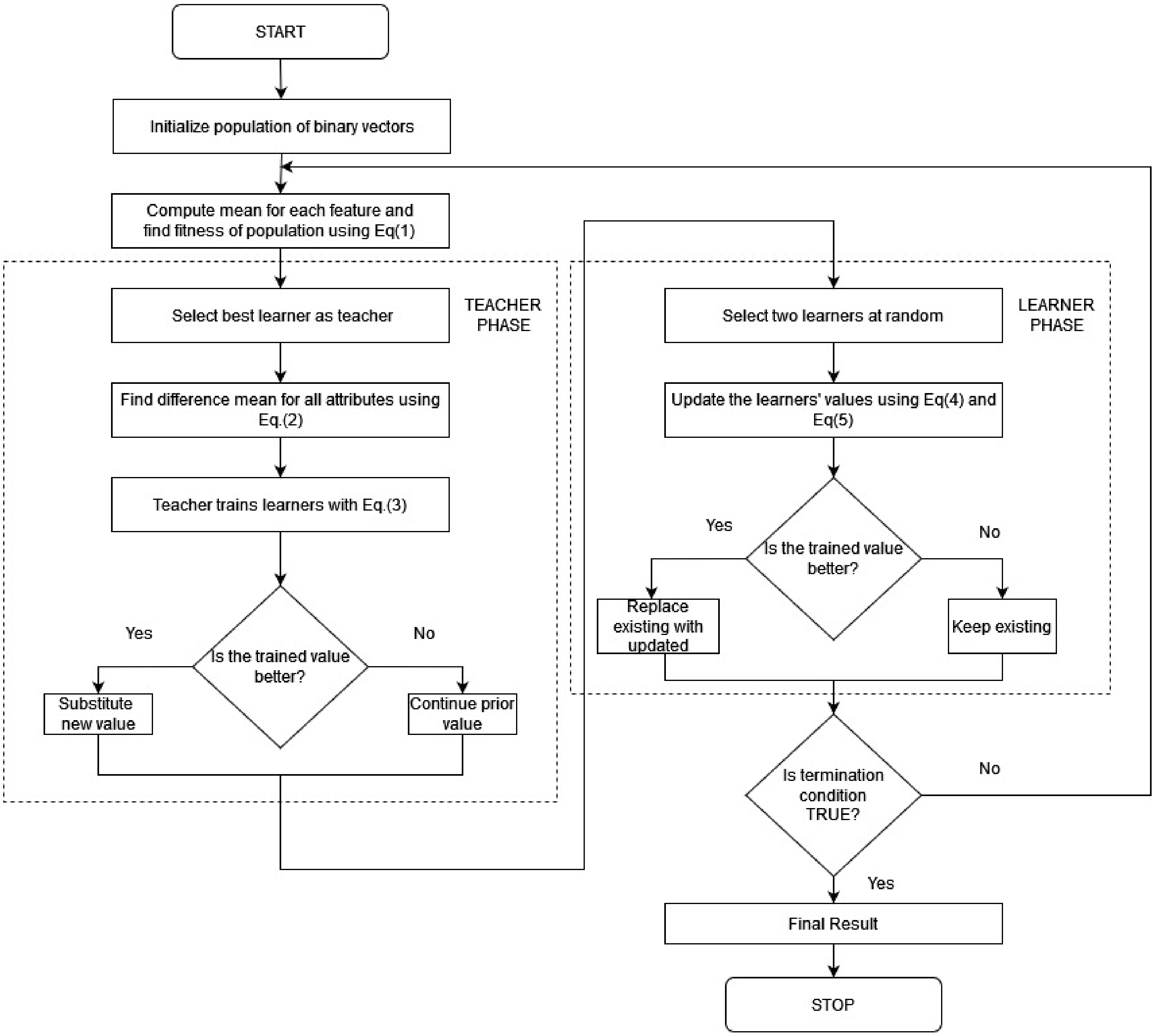

Figure 3.1: Flowchart for TLBO

lutions, and adding it to a third solution. The weights used in this process are usually determined randomly or based on some heuristic criterion.

DE can be used for channel selection by representing each candidate solution as a binary vector, where each bit corresponds to a channel and indicates whether that feature is selected (1) or not selected (0). The goal of the optimization is to find the binary vector that maximizes the performance of a machine learning model trained on the selected channels.

To use DE for channel selection, we need to define an objective function that measures the performance of the machine learning model. This objective function can be a standard performance metric such as accuracy, F1-score, or area under the ROC curve, depending on the task at hand. Appendix I provides a detailed desciption of the formulae used in Differential Evolution

3.3 Causal Convolution

Causal convolution[11] is a type of convolution that is used to filter time series data in a way that preserves the causality of the data. The output at any time step is solely determined by the inputs received up to that point in time, and does not take into account any inputs that may be received in the future. This is important in many applications, such as speech recognition, image processing, and financial forecasting, where it is necessary to predict future values based on past observations.

The term "causal filtering" refers to the fact that the filter only retains "causal" information, meaning information that is relevant to predicting future values of the signal. In other words, the filter only considers past inputs that could have influenced the current output, and ignores any future inputs that have not yet occurred.

Causal convolution can be implemented using a variety of techniques, such as recursive filtering, moving averages, and digital signal processing algorithms. The key characteristic of these techniques is that they only use past inputs to compute the current output, which ensures that the resulting filter is causal.

One important benefit of causal convolution is that it can help avoid the problem of "data leakage" in time series analysis. Data leakage occurs when future information is inadvertently included in the analysis, which can lead to biased results and poor predictive performance. By using a causal filter, we can ensure that only past information is used to predict future values, which can improve the accuracy and robustness of our predictions.

3.4 Dilation

In the context of Temporal Convolutional Networks (TCNs), dilation refers to the spacing between the filter coefficients in a 1D convolutional layer.

In a standard 1D convolutional layer, the filter coefficients are applied to adjacent time steps. For example, a filter of size 3 would be applied to time steps t, t+1, and t+2.

However, in a dilated convolutional layer, the filter coefficients are spaced out by a factor of d, where d is referred to as the dilation factor. For example, a filter of size 3 with a dilation factor of 2 would be applied to time steps t, t+2, and t+4.

Dilated convolutions can be useful in TCNs because they allow the network to have a larger receptive field (i.e., the range

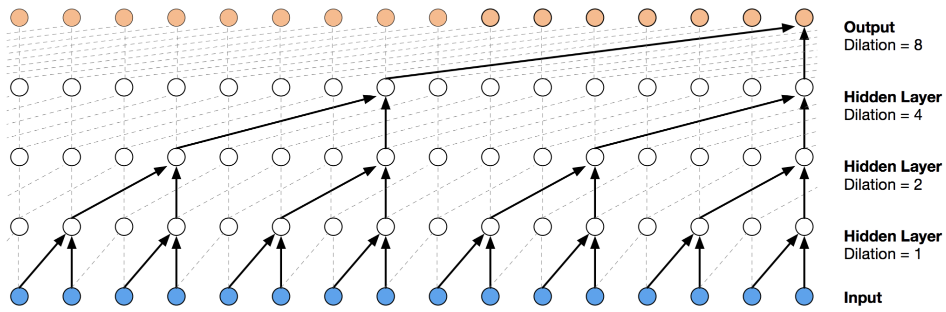

Figure 3.2: A sample TCN model

of input values that affect the output of a particular neuron) without increasing the number of parameters in the network. Such an approach aids the neural network in comprehending long-term relationships among the inputs within the sequence, making it particularly valuable for tasks such as speech recognition and language modeling.

Chapter 4

Related Work

As EEG produces signals as data, a large amount of preprocessing is required to utilize the data. Upon closer examination of EEG, it becomes evident that non-invasive brain-computer interfaces (BCI) can be distinguished based on the techniques used to monitor brain activity and convert various brain signals into commands that can control an effector. The EEG dataset involving motor imagery[2] has made significant contributions to the field of EEG signal processing, and has facilitated the development of several models for analyzing and interpreting EEG signals.

In the real world EEG continues to play an extensive role in the diagnosis and management of patients with seizure disorders like epilepsy. Researchers have also created the concealed information test (CIT)[5], which offers a quick and accurate approach to determine whether someone holds any concealed knowledge, using the P300 component of event-related potential (ERP). This is a more effective method of lie detection[4] than the polygraph tests that are used in the current environment.

This is why implementing Deep Learning models for EEG datasets is crucial these techniques automatically extract a more universal feature set that may be used for a variety of applications.

The use of Recurrent Neural Networks (RNNs) has been extensively explored in the context of the Auditory and Visual Rhythm Omission EEG dataset, as described in studies conducted by Sweet et. Al[14].

Furthermore, TLBO has been implemented for optimal channel selection in EEG-based emotion recognition[9], which involves identifying the emotional state of an individual based on their brainwave patterns.

There have also been studies conducted on several modified DE algorithms that work by adaptively adjusting the mutation and crossover rates based on the fitness of the solutions, which allows for more effective exploration and exploitation of the solution space. These algorithms have been shown to achieve promising results in EEG signal denoising, with improved classification accuracy compared to other denoising methods, such as wavelet denoising and Principal Component Analysis (PCA).

Chapter 5

Proposed Solution

The main objective of our work done thus far was to implement meta heuristic algorithms like DE and TLBO. While both DE and TLBO have been shown to be effective for channel selection in various applications, they also have their limitations. To overcome these limitations, we proposed a hybrid algorithm that combines the strengths of both DE and TLBO. Our approach leverages the exploratory power of DE and the exploitation power of TLBO to enhance the efficiency of channel selection.

The proposed algorithm improves the performance of existing channel selection methods, especially for EEG channel selection and is also an alternative to other methods of channel selection like Independent Component Analysis (ICA), Principal Component Analysis (PCA), and many more statistical and mathematical methods.

After channel selection, temporal convolutional networks or TCNs have been used for classification. TCNs take into consideration both the temporal and spatial aspects and characteristics of the EEG signals and their results have been compared against LSTM classifiers. TCNs have proved that they can massively outperform LSTM networks by using the concepts of causal convolutions and dilations.

In summary, our work has demonstrated the potential of a hybrid algorithm that combines DE and TLBO for channel selection in EEG classification tasks. The proposed algorithm can be applied to other classification tasks and can potentially improve the performance of existing channel selection methods. The entire proposed model can be seen in Figure 5.1.

5.1 Data Preprocessing

EEG data must be preprocessed and examined after data acquisition. Pre- processing entails a number of procedures intended to increase the data's signal-to-noise ratio and make it easier to spot any experimental effects that may be present.

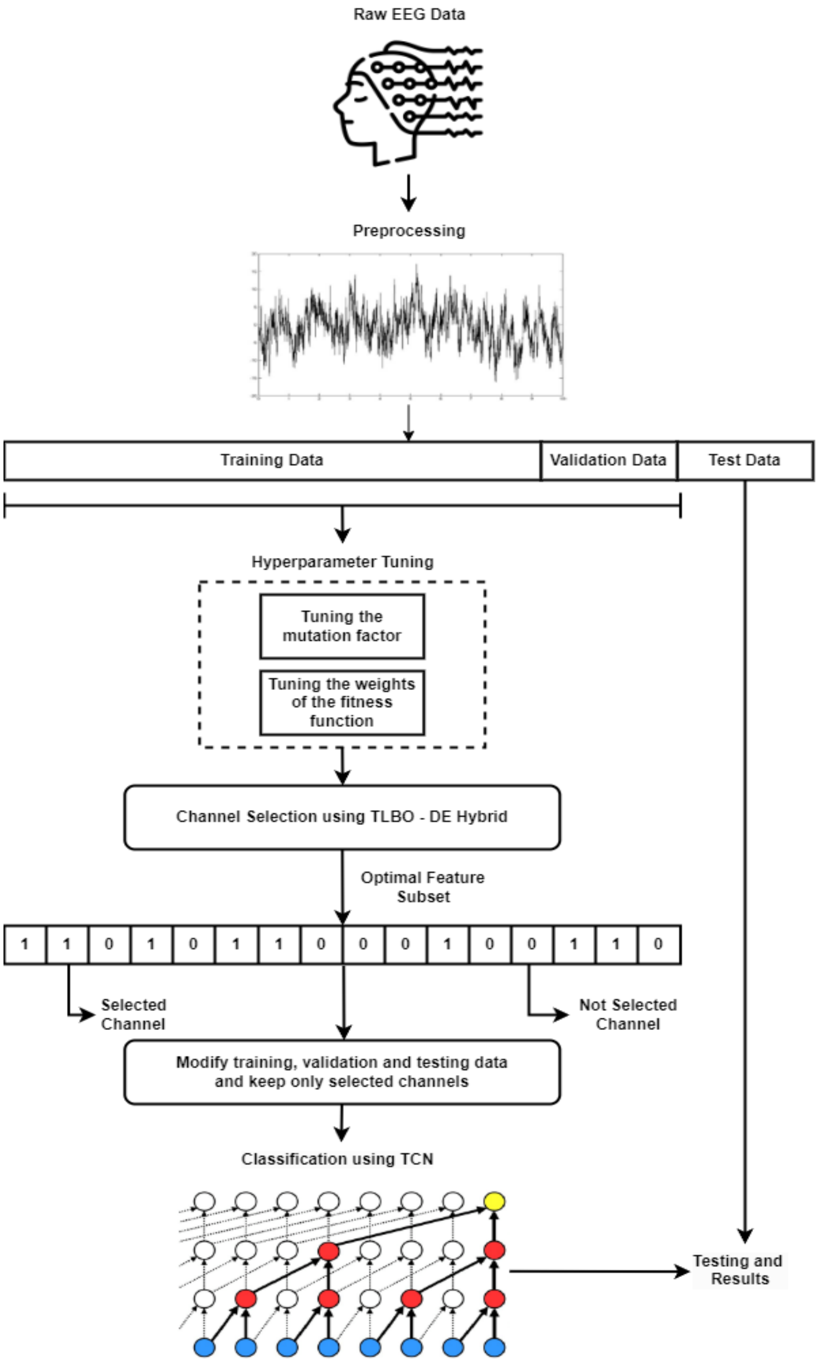

Figure 5.1: Proposed Model Flowchart

The raw EEG data is first standardized using StandardScaler(). The events data is then looped through and then intervals of a fixed length are sliced from the raw EEG data. The final dataframe is built by stacking the resultant arrays into a single three-dimensional array where the 3 dimensions are **total number of trials**, **duration or length of each trial** and **number of channels**.

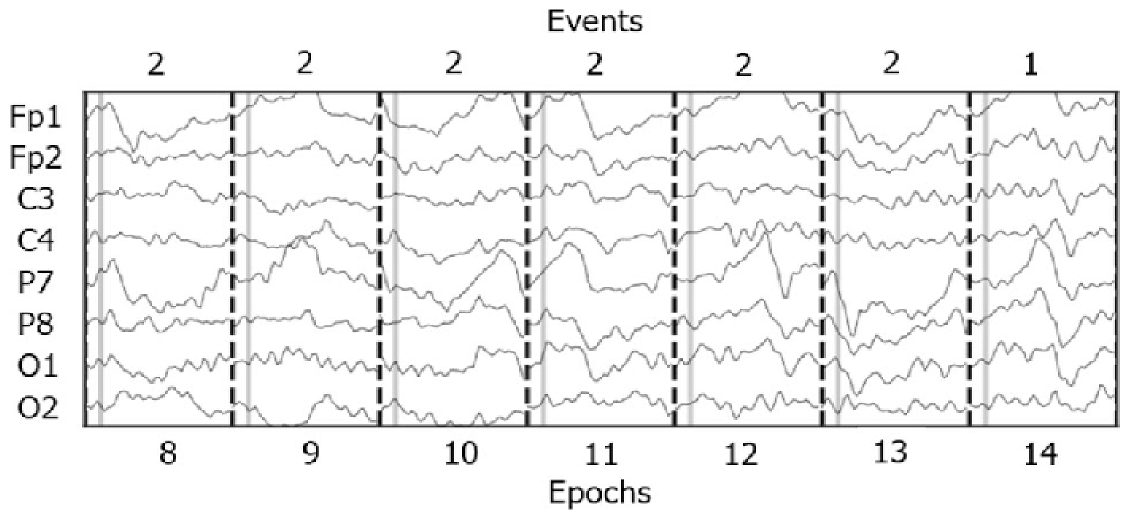

Figure 5.2: Segmenting the data into epochs

5.2 Channel Selection

The EEG data that is obtained after preprocessing will consist of many channels where some might contribute to noise and hence affect the model and classification negatively.

We have used the binary TLBO-DE hybrid algorithm to select the optimal subset of channels from the EEG data which will help pick channels that give us a better accuracy and filter out and discard the EEG channels that add noise.

The TLBO-DE hybrid algorithm that we have developed can be seen in Algorithm 1. The major modifications include adding mutation provided by differential evolution to the TLBO algorithm which originally consists of only the teaching and learning phase. This helps the algorithm to converge faster by generating a more diverse set of individuals. The mutant individual is calculated by selecting 4 random individuals from the population and using the formula

$$\text{mutant} = a + F \cdot (\text{mean} - b) + F \cdot (d - e) \tag{5.1}$$

This proves to be better than just randomly flipping one of the bits in the binary individual as it considers the difference with the mean and the difference between 2 individuals. The mutant is then clipped to keep it within the bounds and its fitness is calculated. The input to the TLBO-DE algorithm is the training dataset, the validation dataset, the population size, number of iterations and the DE mutation factor (hypertuned by trial and error). After the initial population is created, fitness of each individual is calculated using the fitness function. The fitness function used in the algorithm is a **multi-objective fitness function** and consists of 2 objectives which are **increasing the accuracy** of the classifier and simultaneously **reducing the number of channels** used.

The fitness function used is as follows:

$$fitness(individual) = w_1 \left[1 - \frac{\text{No. of channels used}}{\text{Total No. of channels}} \right] + w_2 \left[\frac{Accuracy}{100} \right] \tag{5.2}$$

The accuracy can be any metric used by the classifier like accuracy on the validation data, F1 Score, etc. The accuracy of an LSTM classifier on the validation dataset is used here.

The two objectives are given weights w_1 and w_2 respectively. The weights are assigned based on the task at hand. As accuracy is more important than number of channels used it is given a higher weight. Therefore, the weights were hypertuned by varying weight w_1 from 0.1 to 0.5 and weight w_2 from 0.5 to 0.9. The optimum values which gave a high fitness as well as satisfied the problem statement were found to be $\mathbf{w_1 = 0.2}$ and $\mathbf{w_2 = 0.8}$.

Next, the mutation factor F is hypertuned to pick the optimum value to be used for channel selection. Channel selection also reduces the dimensionality and size of the dataset which makes it easier and faster to train a model on the dataset.

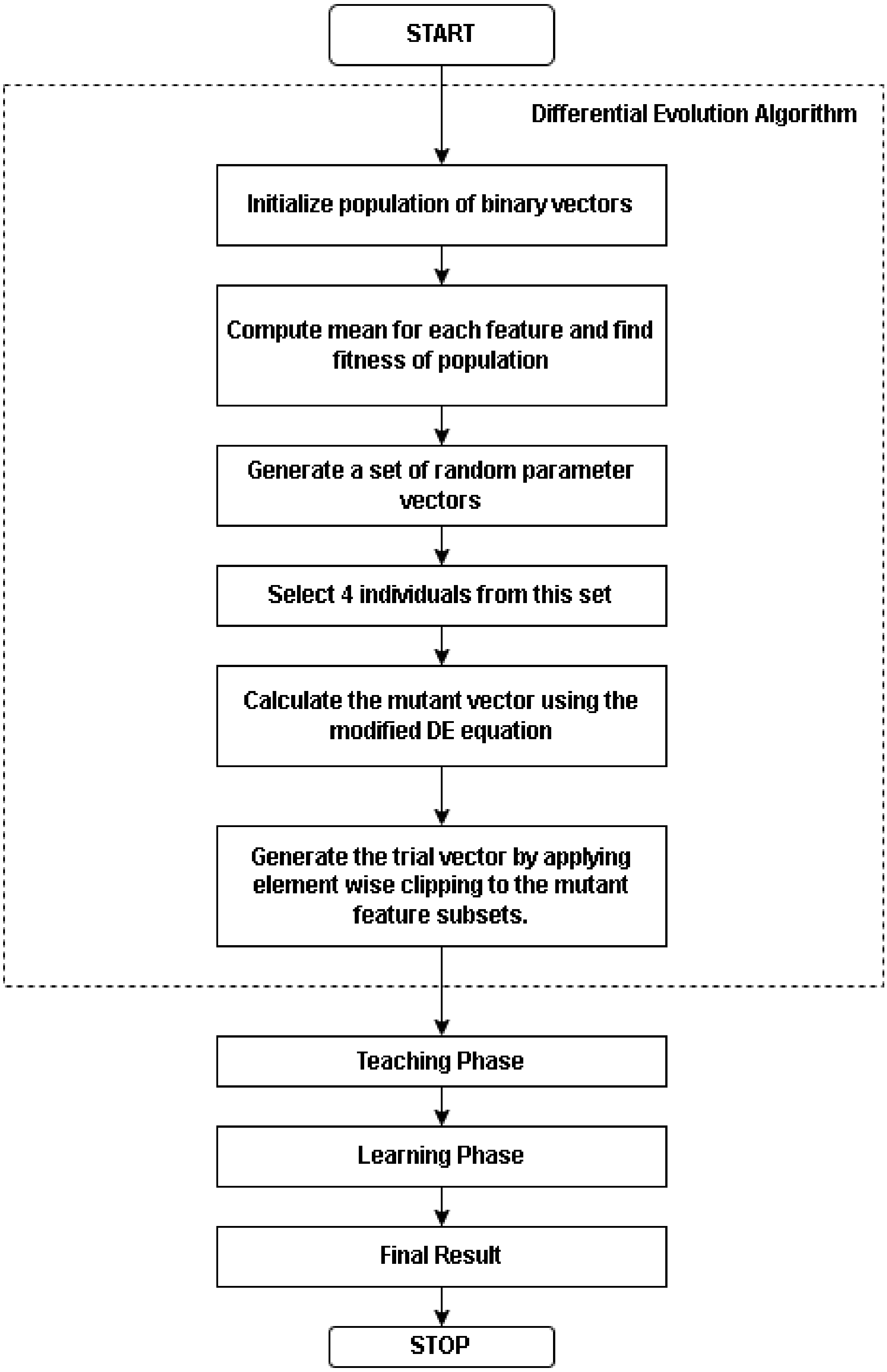

Figure 5.3: Flowchart for TLBO with Differential Evolution

Algorithm 1 TLBO-DE Channel Selection Algorithm

Require: Objective function, bounds, population size pop_size, maximum number of iterations max_iter, scaling factor F
Ensure: Best solution found
Initialize population: Generate a population of binary vectors (0's and 1's) randomly within the given bounds.
Evaluate the fitness of each vector using the objective function.
for $i \leftarrow 1$ to max_iter **do**
    Calculate the mean of the population.
    Randomly select 4 individuals from the population (a, b, d, e).
    Generate a mutant vector using a, mean, d and e as follows:
        $mutant = a + F * (mean - b) + F * (d - e)$
    Clip the values of mutant vector to ensure they are within bounds.
    Create a trial vector by combining the mutant vector and a randomly selected individual from the population:
    Evaluate the fitness of the trial vector.
    Replace the worst individual in the population with the trial vector if its fitness is better than that of the worst individual.
    Recalculate the mean of the population.
    Perform Teaching-Learning-Based Optimization (TLBO) on the population:
    for $j \leftarrow 1$ to pop_size **do**
        Find the best and second best individuals in the population (excluding the current individual).
        Calculate the difference vector between the current individual and the mean.
        Generate a new individual during the teaching phase:
            $new\_individual = current\_individual + random\_uniform() * (best\_individual - F * difference\_vector)$
        Learning Phase:
            $new\_individual = new\_individual + random\_uniform() * (second\_best\_individual - F * difference\_vector)$
        Clip the values of $new\_individual$ to ensure they are within bounds.
        Evaluate the fitness of the new individual.
        if $fitness(new\_individual) > fitness(current\_individual)$ **then**
            Replace the current individual with the new individual.
        end if
    end for
end for
Return the binary vector with the highest fitness as the best solution.

5.3 Tools used

5.3.1 Tensorflow

TensorFlow[8] is an open-source machine learning framework developed by Google that is widely used for building and deploying machine learning models. It provides a flexible and efficient programming interface for building and training various types of deep learning models, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs)[1], and transformers. TensorFlow represent the computations in models using dataflow graphs, which allows for efficient execution across multiple devices such as CPUs and GPUs. TensorFlow can be used in Python, making it accessible to developers who are familiar with the language, and it also supports other programming languages like C++, Java, and Go.

5.4 Methods

Here we have briefly described the various methods and models we have implemented on the datasets.

5.4.1 TCN Implementation

TCN stands for Temporal Convolutional Networks[10], a neural network architecture used primarily for processing sequential data like time-series data.

The key feature of TCN is the use of 1D convolutions with dilations, which allows for a much larger receptive field than traditional convolutional neural networks. This means that TCN can capture long-range dependencies in the data, making it well-suited for tasks that require modeling sequences with complex dependencies.

TCN has several advantages over traditional RNNs like LSTM and GRUs. For instance, TCN is parallelizable and computationally efficient, making it much faster to train than RNNs. Additionally, TCN does not suffer from the vanishing gradient problem, which is a common issue with RNNs that can make it difficult to train them on long sequences.

Overall, TCN has shown promising results on various sequential data tasks, and it is a popular choice for researchers and practitioners in the field of deep learning.

5.4.2 LSTM Implementation

In this hybrid approach for channel selection, LSTM (Long Short-Term Memory)[7] is used in the fitness function as well as for classification of the data. LSTM, like RNNs, work with sequential data and is used in various fields, including natural language processing, speech recognition, and time-series analysis.

In this approach, the LSTM extracts important channels from the input , these channels are then used as input for the TLBO and differential evolution algorithms. These algorithms select the most relevant channels from the input data, which are later used as input for the LSTM classifier.

The LSTM classifier is trained on the selected channels and predicta the class labels of the input data. The fitness function used in the optimization process is based on the performance of the LSTM classifier, which is evaluated using several metrics.

Overall, this hybrid approach combines the strengths of TLBO and differential evolution for channel selection, with the power of LSTM for sequence modeling and classification. By using LSTM in both the fitness function and classification, the approach is able to better capture the temporal dependencies in the data, leading to improved performance in classification tasks.

Chapter 6

Experiments

6.1 Results on the Auditory and Visual Rhythm Omission EEG dataset

6.1.1 Dataset Description

The Auditory and Visual Rhythm Omission EEG dataset[6] is a dataset of electroencephalogram (EEG) recordings that were collected from human subjects while they were presented with auditory and visual rhythmic stimuli. The aim of collecting this dataset was to investigate the neural mechanisms that underlie auditory and visual processing, as well as the effects of rhythmic stimulus presentation on these processes.

During the rhythmic stimulus presentation paradigm, subjects were presented with a sequence of either auditory or visual stimuli that were presented in a rhythmic pattern with regular intervals between them. In some trials, a stimulus was omitted from the sequence, creating a mismatch or deviation from the expected pattern. These deviations are known to elicit a neural response in the brain called the mismatch negativity (MMN), which is thought to reflect the brain's automatic detection of unexpected or deviant stimuli. This dataset was previously evaluated by Sweet et al. [14], who achieved an accuracy range of 80-86% .

The EEG recordings in this dataset were collected from 32 electrodes placed on the scalp of the subjects, allowing for the measurement of electrical activity in different areas of the brain. The dataset includes both the raw EEG data and preprocessed data, such as event-related potentials (ERPs) and power spectral density (PSD) estimates, which can be used to investigate neural activity in response to the rhythmic stimuli and deviations.

6.1.2 Preprocessing

The raw data was first loaded and it consisted of the raw EEG files and the corresponding event files. This raw EEG data is first standardized using StandardScaler function from the sklearn module. The events data is then iterated over and the corresponding EEG data is sliced out into epochs of length 1.3 seconds and the bad events are filtered out.

The final processed three-dimensional dataset is then built by stacking the epochs to get a dataset of dimensions 3500 x 1300 x 32 which indicates that there are 3500 trials of length 1.3 seconds each with 32 channels.

6.1.3 Training and Evaluation

The dataset was split into training, testing and validation datasets following a 70-15-15% split.

Next, the mutation factor F was hypertuned and this was done by varying it from 0.1 to 1.0 and observing the average fitness of the population after 10 iterations.

The optimum value of mutation factor observed was 0.9 as seen in Figure 6.1, and the hybrid algorithm was then run for 20 iterations and the optimal feature subset was obtained.

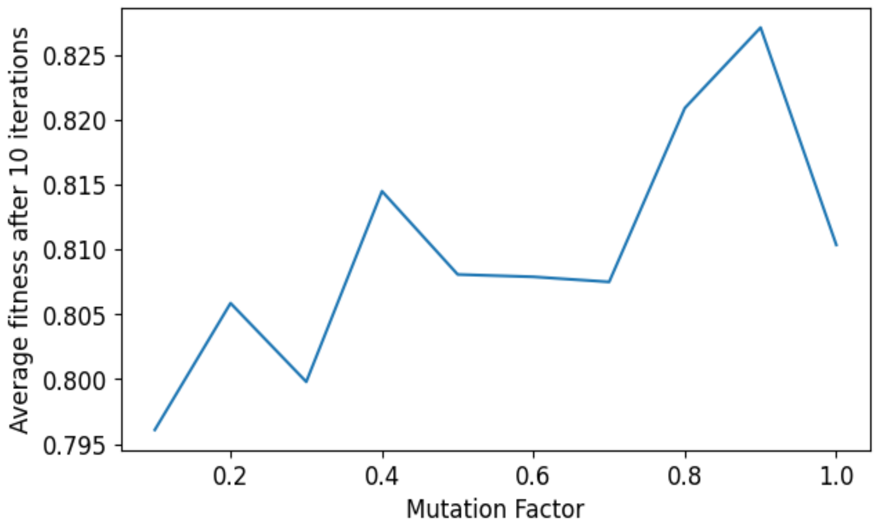

Figure 6.1: Relationship between Mutation Factor and Average Fitness, optimal mutation factor identified at F = 0.9

Channel selection was done using the training and validation datasets using the 3 channel selection methods i.e TLBO, and DE the TLBO-DE hybrid. An LSTM classifier was used in the fitness function to evaluate the fitness of various individuals. Training and testing was done on the dataset before and after applying channel selection and the results were compared using metrics which included the accuracy, precision, recall and F1 score. The training was done using 2 classifiers namely an LSTM classifier and a TCN classifier and the results were compared. The results can be observed in Table 6.1. The TLBO and algorithms picked 17 out of the 32 EEG channels, whereas the

Classifier	Metrics	Before channel selection	TLBO	DE	TLBO-DE
LSTM	Accuracy	71.31±1.45%	71.09±4.71%	72.11±0.53%	75.31±2.78%
	Precision	0.73±0.02	0.78±0.04	0.77±0.05	0.74±0.04
	Recall	0.81±0.03	0.73±0.12	0.77±0.07	0.91±0.04
	F1 Score	0.77±0.02	0.75±0.06	0.77±0.01	0.82±0.01
TCN	Accuracy	94.69±1.04%	93.90±0.77%	93.61±1.04%	96.87±0.54%
	Precision	0.98±0.01	0.98±0.01	0.98±0.01	0.98±0.01
	Recall	0.92±0.02	0.91±0.01	0.91±0.01	0.96±0.02
	F1 Score	0.95±0.01	0.95±0.01	0.94±0.01	0.97±0.01

Table 6.1: Experimental Results of the various algorithms and classifiers

TLBO-DE hybrid picked 19 out of the 32 channels as seen in Table 6.2 and the hybrid algorithm gave an increased accuracy of 96.87% on the testing data when training was done using the TCN classifier which are the best results. The TLBO-DE hybrid has therefore excluded 13 channels resulting a 40.625% decrease of data. Therefore, the TCN classifier applied on the features selected by the hybrid channel selection method gave us the best results.

Channel selection Method	Number of Features Selected
TLBO	17
DE	17
TLBO-DE	19

Table 6.2: Number of channels selected by each method

The receiver operating characteristic curves or ROC curves have been plotted for the three methods showing the performance of the classification model at all classification thresholds. The red, blue and green lines indicate the TLBO, DE, and TLBO+DE hybrid respectively. It is observed that the area under the curve or AUC is the greatest for the TLBO+DE curve and it is greater than that of the other 2 curves therefore outperforming them.

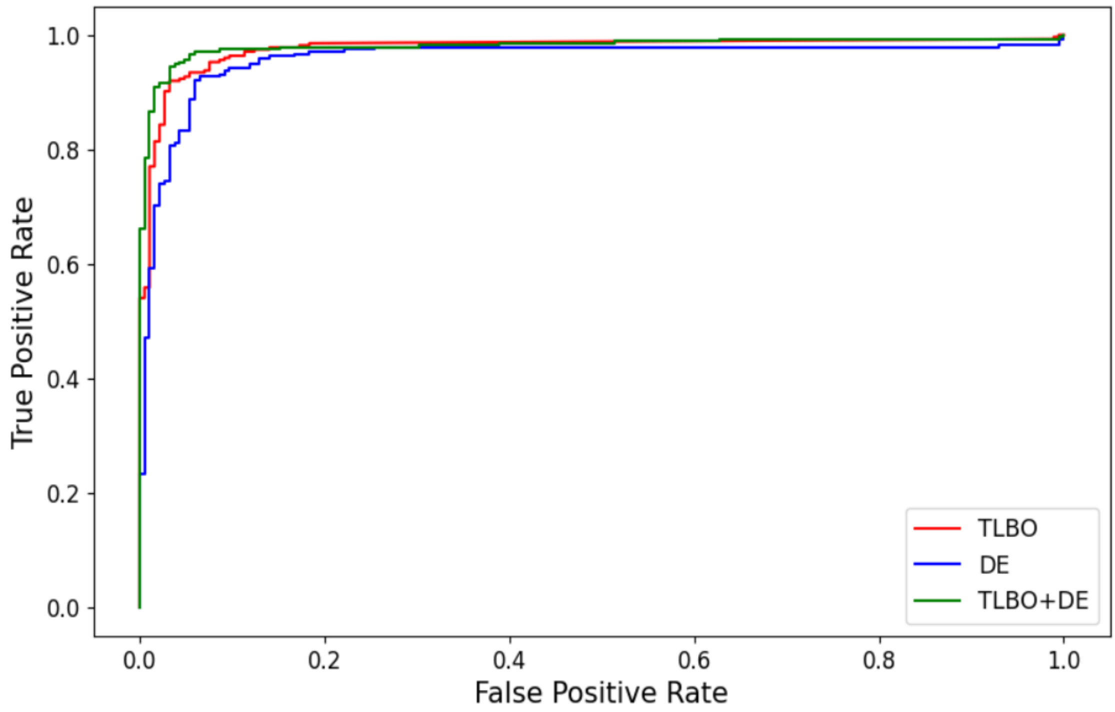

Figure 6.2: ROC Curves for the 3 channel selection methods

The optimal subset of 19 channels included the following channels :- **FP1, FPZ, FP2, F7, F3, F4, F8, FC5, FC2, FC6, T7, CZ, T8, CP1, CP2, P7, P8, POZ & O1**.

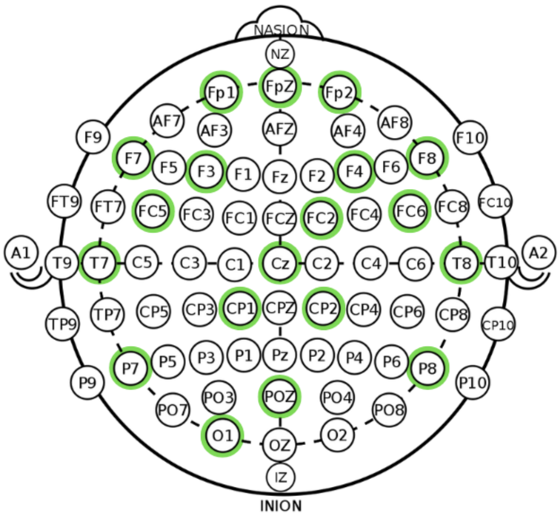

Figure 6.3: Location of channels selected by TLBO+DE algorithm

The location of these 19 electrodes on the scalp of a subject is shown in Figure 6.3. The algorithm has therefore picked channels that are spread out all over the scalp and left out channels that are very close to each other.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Analysis of EEG signals provides a non-invasive means of studying the cognitive behaviors of individuals. By examining the electrical activity in the brain, EEG analysis can reveal insights into cognitive processes such as attention, memory, perception, and decision-making, among others. This is better than manually interpreting these signals as they are time - consuming and expensive. We have used deep learning techniques, taking inspiration from TLBO and DE algorithms, creating our own hybrid of the samec thus automating this process to obtain a considerable degree of accuracy that surpasses that of previous results obtained in other studies.

7.2 Future Work

Our future work will include applying our model to other EEG applications and also extend the use of the hybrid TLBO-DE algorithm to other medical and non-medical domains. This will validate the effectiveness and generalizability of our approach and explore the potential of using EEG data for improving diagnostic accuracy in neurological disorders. Our ultimate goal is to develop a reliable and useful tool to infer and extract important information about the cognitive brain function.

Bibliography

[1] Recurrent neural networks for optimization. *Recurrent Neural Networks*, page 22–54, 1999. 5.3.1

[2] Waseem Abbas. Bci competition 2008 – graz data set a experimental paradigm, Feb 2016. 4

[3] Mohan Allam and M. Nandhini. Optimal feature selection using binary teaching learning based optimization algorithm. *Journal of King Saud University - Computer and Information Sciences*, 34(2):329–341, 2022. 3, 3.1

[4] Annushree Bablani, Damodar Reddy Edla, Venkatnareshbabu Kupilli, and Ramesh Dharavath. Lie detection using fuzzy ensemble approach with novel defuzzification method for classification of eeg signals. *IEEE Transactions on Instrumentation and Measurement*, 70:1–13, 2021. 4

[5] Annushree Bablani, Damodar Reddy Edla, Diwakar Tripathi, and Venkatanareshbabu Kuppili. An efficient concealed information test: Eeg feature extraction and ensemble classification for lie identification. *Machine Vision and Applications*, 30(5):813–832, 2018. 2.1, 4

[6] Daniel C Comstock and Ramesh Balasubramaniam. "auditory and visual rhythm omission eeg", 2019. 6.1.1

[7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 5.4.2

[8] Vikraman Karunanidhi. Understanding tensorflow javascript library. *Deploying TensorFlow Models to a Web Application*, 2020. 5.3.1

[9] Nidhi Kumari and Rohit Bhatia. Efficient facial emotion recognition model using deep convolutional neural network and modified joint trilateral filter. *Soft Computing*, 26(12):7817–7830, 2022. 4

[10] Colin Lea, René Vidal, Austin Reiter, and Gregory D. Hager. Temporal convolutional networks: A unified ap-

proach to action segmentation. *Lecture Notes in Computer Science*, page 47–54, 2016. 5.4.1

[11] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio, Sep 2016. 3.3

[12] R. Venkata Rao. *Teaching learning based optimization algorithm and its engineering applications*. Springer International Publishing, 2016. 2.2, 3.1

[13] Rainer Storn and Kenneth Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces - journal of global optimization. 2.2, 3.2

[14] Tyrome Sweet and Taran Rallings. Recurrent neural network-based approach for classifying audio and visual stimuli eeg data. Dec 2017. 4, 6.1.1