WHAT ARE THE SUITABLE LEARNING ALGORITHMS BASED ON EEG SIGNALS FOR EMOTION DETECTION?

Prithivi Sharma, 45705704

Bachelor of Engineering(Honours)
Software Engineering Stream

School of Engineering
Macquarie University

October 22, 2022

Supervisor: Dr, James Zheng

ACKNOWLEDGMENTS

I would like to acknowledge my thesis supervisor as well as the project team who helped me get familiarised with this project and its contents for me to be able to complete my thesis paper.

STATEMENT OF CANDIDATE

I, Prithivi, declare that this report, submitted as part of the requirement for the award of Bachelor of Engineering in the School of Engineering, Macquarie University, is entirely my own work unless otherwise referenced or acknowledged. This document has not been submitted for qualification or assessment an any academic institution.

Student's Name: Prithivi Sharma

Student's Signature: Prithivi Sharma

Date: 4/11/2022

ABSTRACT

Electroencephalogram (EEG) signals have been an integral area of study in recent times due to its applications in deep-learning models to analyse Brain Computer Interaction (BCI). Due to that and the wide array of different methods that can be applied in enhancing ways to classify human emotions, this paper analyses the different algorithms, conventional and state-of-the-art which have been able to collect and classify human emotions using many different frameworks and algorithms. This includes the use of Machine Learning and Deep Learning models. In addition, this paper also proposes a pilot method which focuses on areas of the brain used to generate a physical reaction based on the stimuli presented to participants of the project to test the current network architecture so that it can be used for future experiments that contain more complex models. This is achieved by focusing on the collection of raw EEG signals and their preprocessing, feature extraction and classification methods, which serve as a model to transition to deep learning methods which have been proven to be more accurate in being able to classify and approximate human emotion based on stimuli presented to participants. The current pilot study applies random forest library as the primary source of data classification with a time domain feature extraction framework that particularly focuses on areas of high frequency within the human brain through the participant's interaction with a given stimuli. This in turn, allows us to identify different aspects of this implementation which can be improved to gain a stronger level of accuracy when implementing different algorithms when looking to classify emotions in the future as evident from its initial accuracy reading of 64% when looking at the small number of participants of 5.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The identification and understanding of human emotion through the use of EEG signals has become an area of significant importance. This is evident from its application on understanding the interaction of people with many sectors such as Machine Learning, Education and health care [6]. From its introduction to its growth in popularity for capturing human emotion, the ability of EEG signals and the algorithms that are used with it have become a strong topic of discussion when looking at how accurate each algorithm is when it comes to successfully classifying and classifying human emotion. This can be attributed to many different factors that are responsible in being able to generate a model that is accurate in classifying human emotion. This varies from the headset used to capture the brain waves to the age and number of participants for the experiment as well as the libraries that are responsible for collecting and processing data that will be used to represent the accuracy of a model. Therefore, this project first looks to analyse traditional machine learning algorithms and state-of-the-art deep learning algorithms through the use of features and formulas they use to generate their accuracies. In addition, by further understanding the different elements involved in calculating the accuracies of each type of model, this paper looks to highlight the advantages and drawbacks each set of algorithms and their respective models produce in the context of using EEG signals to classify human emotion and what can be done to improve the classification accuracies they present.

Along with that, this paper will also look to use the knowledge generated through the different studies of algorithms used to collect and analyse EEG brain wave signals. This study will also analyse the initial pilot method for this project with a time-domain model that applies a Random Forest classifier with 5 participants to classify limb movement of participants through EEG signals from highly concentrated areas of the human brain [3]. By being able to do so, this initial experiment will act as a strong base to expand into different algorithms that can help in significantly improving the accuracy of our model in the future as this project ventures to the application of different, more advanced algorithms to better collect and understand EEG signals and thus, improve its accuracy when looking at a larger scale of participants.

Therefore, this paper looks to discuss different algorithms and models that are used to classify human emotion through the capture and processing of EEG signals. Section 2 provides a literature review of the many different algorithms and their respective models in the past that have been used in classifying human emotions, Section 3 describes the preliminary pilot study that uses a network architecture for emotion classification with raw data from EEG signals. Upon evaluation of said model, it also highlights areas it in

regard to the capture and process of EEG signals that can be improved for the future model that is to be developed to be more accurate.

1.0.1 Background

An Electroencephalogram (EEG) cap is used to collect information from a participant from the electrodes placed on a cap. Although there are many approaches applied in the placements of electrodes, the International standard of 10-20 is implemented as a means to describe the position of electrodes within an EEG cap as shown in as seen in Fig 1.1 [1]. The electrodes are used to look over brain activity of a user by recording signals from the brain. This is done by analysing the brain wave bands that are found in EEG signals that include Delta(1-4), Theta(4-8), Alpha(8-14), Beta(14-31) and Gamma(31-50). EEG caps, then transfer the raw signals to computer systems through different means.

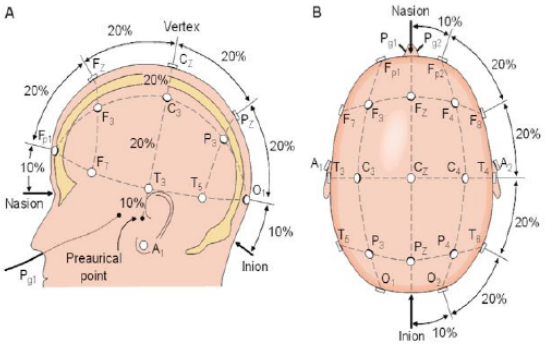

Figure 1.1: 10-20 EEG system

Collection The collection of EEG signals used for processing in models are achieved through some major steps. The first steps include choosing the type of stimuli used to collect brain activity and data from EEG caps. Once decided, the number of participants whose data are collected is chosen. Once chosen, chosen participants interact with the stimuli picked for them and data is collected for processing.

Processing Raw EEG signals collected from all channels are then passed on for pre-processing. Pre-processing involves the removal of unwanted data, which are normally identified as noises. This process also includes the division of data into training and testing data, allowing for data to be clean, resulting in accurate models.

Feature Extraction The stage of feature extraction is used as an area to further process pre-processed data by extracting features through many different algorithms catered to certain elements of the EEG signals. This can range from differential entropy(DE) for temporal features [6, 23] to fast fourier transform(FFT) features. Consequently, this allows for the classifier at the end of the model to be able to extract an emotion based on the signal provided and the functions used to process it with more accuracy.

Classification The classification stage is used to select the classifiers that are used to allocate an emotion based on the number of classes and model selected. This can include valence-arousal which is selected for majority of the projects used for EEG signal processing models for emotion detection.

1.0.2 Related Work

Earlier iterations of models used to collect, process and classify EEG signals for emotion detection mostly centred around the implementation of hand-crafted algorithmic approaches with deep belief networks and machine learning classification libraries such as Support Vector Machines(SVM) and K-Nearest Neighbor(KNN). They had limited capabilities on feature extraction which would lead to increased computational resources in the creation of a classification model resulting in the application of deep learning methods over the years as seen in Fig 1.3 [16].

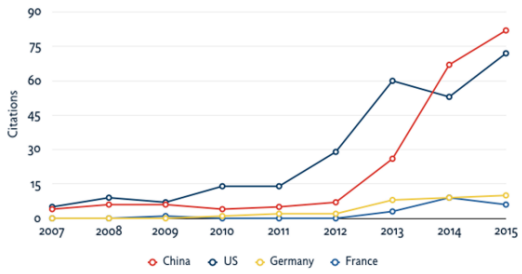

Figure 1.2: Publications about Deep Learning from - China, USA, Germany and France

Deep Belief Network

Wei-Long Zhong et al investigated the use of Deep Belief Networks with EEG signals acquired from 15 participants to classify emotions on the DEAP dataset. [24] The main feature used for extraction involved the application of DE and value comparisons on the left and right directions of the EEG cap through the differential asymmetry and rational asymmetry. Classification done using Restricted Boltzmann Machines to create a Deep Belief Network. Although it generated relatively good scores with useful formulas and focus towards asymmetrical brain activity for accuracy, some practices in its application required more resources to be applied successful. This included the use of Restricted Boltzmann Machines which are known to be more difficult to train and requires weight adjustment [24]. This can be improved with a network architecture that includes formulas that are more robust and capable of handling large data without needing more time to train and classify EEG emotions.

Hao Chao et al applied a Deep Belief Network Conditional Random Field by using AMIGOS, DEAP, SEED and HR-EEG4EMO. Features were extracted using time-domain, frequency domain and time-frequency domain functions such as mean, variance, zero-crossing rate and power-spectral density [4]. Although the DBNCRF model applies useful features extraction methods based on frequency band placement and time based signals, its accuracy can improve by focusing on using samples from high frequency areas and by

using classification features from state-of-the art deep learning algorithms especially with large scale data [4].

Mohammad Mehdi Hassan et al proposed a deep belief network with an implementation of the physiological signals of Electro Dermal Activity(EDA) along with Photoplethysomogram(PPG) and Zygomatics Electromyography(zEMG) and a Deep Belief Network to be able to find depth level features. The observations of the 3 physiological signals were determined using the principles of the DEAP dataset. The DBN is structured and trained through the successive training of the 3 Restricted Boltzmann machines and sigmoid belief network [9]. Feature extraction is achieved with the 9 statistical features of EDA,PPG and zEMG, 9 Power Spectral Density features of EDA, PPG and zEMG and 46 features of DBN and are then classified using a Fine Gaussian SVM classifier. Although this network focuses on ensuring high frequency EEG data is used for processing and classification through a high pass filter, making more accurate assumptions, it also overlooks the increased training time required for the RBMs along with the classifier due to the large data it is inheriting, resulting in increased computational resources to train the model and generate an accurate assumption. One of the ways to tackle this, however, can be through the implementation of certified state-of-the art deep learning algorithms [6, 18]to minimise the likelihood of errors with built-in and reliable feature extraction and classification algorithms which can allow the model to have a high level of accuracy while also handling large amounts of data at a better rate.

Support Vector Machines

Fabian Parsia George et al implements an SVM classifier with the standard collection, processing and classification procedure for EEG signals with the DEAP dataset for music videos. Band extraction and feature extraction is achieved by the application of FFT and segregated into 4 values of emotional feelings based on valence and arousal, High Arousal- High Valence(HAHV), Low Arousal- High Valence(LAHV), Low Arousal- Low Valence(LALV) and High Arousal- Low Valence(HALV). Classification is achieved by the use of 10-fold cross validation which takes in the scaled statistical features of the input signals for each of the 4 emotional quadrants [8]. Although this experiment was successful in the usage of efficient feature algorithms which are well known for achieving accurate classifications, this model is also hard to maintain due to the inability of the SVM classifiers being able to handle large-scale data as evident from the action taken to reduce the channels that are related to the frontal lobe of the brain. Moreover, the manipulation of the data provided along with all important channels not being used for processing can lead to the model not being a reliable body for emotion classification for EEG signals. This can be improved by moving to a classification model which is able to use feature extraction methods that can cover all areas and handle long sequential data which is essential for this project [6, 23].

Itsara Wichakam et al used a similar implementation process as [8] with an SVM classifier but with multiple normalisation techniques. However, some of the processing steps taken for their model are different due to their stimuli handling 20 songs. The signal processing involved self-assessment after every song was played from the participants which would directly be represented using the Self-Assessment Manikin(SAM) to show the different scales of emotion for valence and arousal with the data collection process focusing on the training of users on how the experiment is conducted and then being tested two times within 5 separate days. The EEG data collected was then passed for feature extraction

methods involved FFT and PSD. The normalisation techniques used for the features extracted included Re-scaling, Z-score standardisation and Frequency Band Percentage with classification done by comparing the performance of the model on the 2 separate tests done through F1 score. Although this model made good use of feature extraction and normalisation methods to pick out high frequency EEG signal areas and process them as seen from its effort to pick the best normalisation technique based on the information provided by using multiple SVM kernels [22], one of the areas this project can be reflected on is of using more pre-processing techniques to remove noise and be able to input signals of strong frequency levels only [15]. One of the areas this project did introduce me to is multiple sessions of testing as it proved to be useful in generating better signals to process in the main network.

Noppadon Jatpaiboo et al on the other hand, propose an EEG signal processing network using SVM with less EEG channels and frequency bands with an SVM classifier achieving a high accuracy. Pre-processing involves choosing 100 stimuli based on the highest and lowest scores on valence within the Geneva Affective Picture Database and Artifact filtering for removing unwanted data. EMOTIV caps and dataset are used in the data collection phase, with 14 channels being used to collect EEG signals. Participants are shown all pictures chosen as their stimuli for 10 seconds with a 5 second time period to adjust their emotion after one picture has been shown and the process continues afterwards. Features are extracted to the 5 frequency bands of Delta, Theta, Alpha, Beta and Gamma through Wavelet Transform(WT [7, 10]. Normalisation of the features from the 14 channels are done using scaling between 0 and 1 and classification is achieved using Gaussian SVM with 10-fold cross validation. The strong points highlighted with this model involve the use of less channels to still generate a strong level of accuracy while also implementing strong pre-processing techniques such as Blind Source Separation to remove unwanted information so that sequences that can be processed to generate a model with strong accuracy is created. Moreover, other practices to improve upon the generated accuracy like reducing pairs of channels and frequency bands also assisted in the learning aspect of this project as it supported how pairing channels close to one another can generate sequential data that is a lot stronger and can give more information on the model itself [6]. An area where improvements can be made is by making comparisons to other sections within the human brain which this project has overlooked which can allow the model to learn more about the relationships between other channels and create pairs of channels that can make the model more accurate than it already is [11].

K-Nearest Neighbor

Mi Li et al proposed a multichannel EEG signal emotion recognition model using the K-Nearest Neighbor classification model [17]. Data collection involves using the DEAP dataset with 32 datasets and 40 videos of 60 seconds. EEG data from 10, 14, 18 and 32 channels from the left and right side of the EEG cap are selected for pre-processing which is done using Average Mean Reference(AMR) and normalised using min-max before the extraction of features. Feature extraction for this model involves the use of Discrete Wavelet Transform(DWT) and features for every channel are generated through Entropy and Energy. The value of K was set to 3 for classification after the extraction of important features. Emotions were classified using valence and arousal dimensions for the 32 participants of the experiment. *Mi Li et al* practice established methods for the

analysis of EEG signals like the very well known Entropy and energy function [12, 18]but does contain some elements that can be improved in order to generate a stronger level of accuracy. One such example is to focus more on a different type of model for classification mainly due to the computational resources and effort to keep a model like operating efficiently using a KNN classifier as data increases in size.

Fatemeh Bahari et al propose a network that uses a KNN classifier and recurrence plot analysis for an emotion detection model using EEG signals. Pre-processing actions included SAM being filled out by the 32 participants based on music and videos after displaying one of 40 music videos for one minute [2]. EEG signals were recorded using 512 Hz and downsampled to 128 Hz. Noise removal was conducted using bandpass frequency filter. Feature extraction involve the use of Recurrence Quantification Analysis and Recurrence Plot Analysis. Classification was achieved through the use of methods such as t-test and area under ROC curve. Case independent classification involved the exclusion of the first 10 sets of features randomly and the use of statistical methods and the Bhattacharyya distance. KNN and valence, arousal and liking were used as classifiers for the final classification. Case dependent classification was also implemented across all channels using a LOO procedure and included same optimal values for k values and distance metrics. Some areas which were responsible for the low accuracy generated from this model was the improper use of the channels during classification as the exclusion of extraction of temporal features as this model focused only on spectral features of EEG. This is further supported from the lack of research that was conducted on automatic emotion recognition using EEG [2]. Model accuracy can be improved by using different classifiers and the inclusion of other extraction methods that have been known to be effective in emotion recognition using EEG signals [18].

Kaundaya et al breaks down an EEG signal emotion detection model with a KNN classifier. Pre-processing involves the collection of EEG signals from ground truth from visual and audio stimuli. Band pass filter is used to remove noises of 50Hz and the DC offset from each electrode of an EEG cap [14]. Feature extraction process involves statistical parameters calculation using WT and coefficients for four level decomposition are extracted using mean, Variance, standard deviation, entropy, root mean square, skewness and power [7]. KNN classification model is uses the new labeled sample passed through as testing data with the baseline data during preprocessing as the training data. The classifier is evaluated from the varied value of k from 1 to 10. Accuracy is obtained from the values of the total datasets that are selected for the model. Although this model is thorough with classification analysis through the examination of different k values from the KNN classifier, increased datasets can prove to be a challenge in terms of maintaining the accuracy of the model. Moreover, feature extraction methods like Linear Discriminant analysis can prove to be more efficient for this model as it allows for more maximisation within mean values of each class and can lead to less overlap between classes which is essential for KNN classifiers, therefore, improving its accuracy.

Chapter 2

Deep Learning Algorithms To Classify Human Emotion

2.1 State Of The Art Algorithms (Deep Learning)

State-Of-The-Art Deep Learning Algorithms are a strong method in identifying patterns in computer vision and have been heavily used in image classification. Some recent studies implementing these models [6, 18, 23] have been in-part successful in detecting human emotion through EEG signals. Whereas CNN has been successful in image processing, RNN is useful in speech recognition, language translations as well as for image processing which is what is used for EEG signals.

2.2 Convolutional Neural Networks (CNN)

A convolutional neural network is a feed forward, deep learning neural network that is applied to process structured pieces of information. These include images as well natural language processing for text classification. They are widely known to be a state-of-the-art neural network when it comes to computer vision and image classification.

2.2.1 Pre-Processing

Heng Cui et al uses DEAP and DREAMER datasets [18,19] to get rid of basic emotional states that participants have without being exposed to any form of stimuli. This is done by dividing the trial data of 60 seconds and baseline data of 3 seconds into individual samples of each trial with the remaining baseline signals captured being represented as the basic emotional state of the participants. Once found, the basic emotional states and their effect on the main EEG signal are removed from the current captured EEG signal through the deduction of each trial sample which was calculated before with the mean value of the same trial [6]. Once completed, Z-score is used in order to normalise the data based on the EEG channels and the time where the signals are captured through the function of: $x^N \epsilon R^{cxs}$.

In comparison, the use of DEAP datasets within the preprocessing phase by *Tran-Dac-Thinh-Phan et al* is used as an indicator of the effectiveness of their model and EEG signals are captured within trials of 60 seconds with a given stimuli whereas the process

of improving the quality of the captured signal is done by downsampling the given signal to 128 Hz to find sections of low arousal and negative valence which was then filtered using a band-pass filter with cut-off frequencies from 4 to 45, which focused on the frequency bands of Theta, Alpha, Beta and Gamma [18]. Once cut off, the Butterworth filter [15] of 30 is chosen to remove electromagnetic interferences and noises and to make no changes to the captured signal while bands are overlapping with another. Afterwards, sliding window is used to divide one main signal of 60 seconds into 10 different segments.

Lili Shen et al, although applies the same principal of each signal trial being 60 seconds and baseline signals of users being 3 seconds like other experiments that aforementioned, it however, focuses more on the capture of temporal and spatial sections of EEG signals through layer normalisation. This allows for the network to be able to find hidden layers of representation within captured EEG signals to be processed. A baseline noise filtering module is used to remove baseline signals with fluctuation [20]. Moreover, a SAM score [12] is done by participants on each video which is transformed to high or low valence or arousal based on the score of greater than or equal to 5 or less than 5 which i added on as data as well. [20]

Elham S. Salama et al used an extension of the conventional convolutional neural network with a face-based input used in emotion recognition combined alongside EEG signals which are captured without focusing towards frequency bands but with all channels combined and 5 consecutive frames, creating an input chunk for EEG signals [19]. This would include the temporal domain data which is gathered from the frames and added alongside the EEG chunk. Face based emotion recognition is achieved with the input video of the user being reduced to 30 frames per second and only 60 frames being selected. The quality of the input data is increased through changes in brightness, updating the color and flipping the image taken left to right. In addition to that, the **Mask-RCNN** is applied to the input face frame and assigns a class for each object it identifies. Once assigned, OpenCV is implemented to extract pixels from the image used to identify faces, resulting in the 5 frames which will be used as the main input for the main network, generating a face chunk [19].

2.2.2 Feature Extractor and Classification

2D CNN

Tran-Dac-Thinh-Phan et al highlight this from the multi-extraction framework they introduce with a focus toward time-domain features which was used to improve emotion classification of the signals presented. This framework involved the use of 4 matrices representing each frequency band as shown in Fig 2.1 [18]:

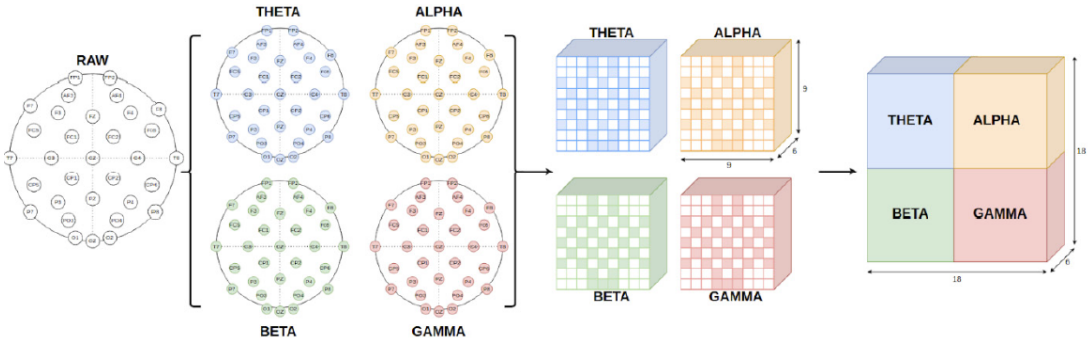

Figure 2.1: Feature extraction framework for 2D CNN

Focusing on each individual segment which was made during preprocessing in order to improve emotion classification. This would be achieved using features like DE(1), mean(2), mean of the first difference(3), mean of the second difference(4), variance(5) and standard deviation(6).

$$DE(X) = -\int_{-\infty}^{+\infty} p(x)log(p(x))dx \qquad (1)$$

with X representing the sequence of random variables that the segments have been broken down to and probability density function(PDE): **p(x)** being applied to the segment where it considers all values of the sequence and determines the probability of those values occurring throughout the segment.

$$\overline{\mu} = \frac{1}{n}(\sum_{i=1}^{n} x_i) \qquad (2)$$

with n representing the number of segments that the main signal has been broken down to, generating an average value which indicates the RGB properties of every random variable within the sequence as well as its channels.

$$\overline{\mu_1} = \frac{1}{n-1}(\sum_{i=1}^{n-1} x_{i+1} - x_i) \qquad (3)$$

$$\overline{\mu_2} = \frac{1}{n-2}(\sum_{i=1}^{n-2}(x_{i+2} - x_i)) \qquad (4)$$

These formulas are used as pattern recognition methods by finding the difference between the values of the initial random variable that was chosen. This can lead to the discovery

of linear correlation, satisfying one of the aims of the selected feature extraction methods [18].

$$\sigma = \sqrt{\frac{1}{n}(\sum_{i=1}^{n}(x_i - \overline{\mu})^2)} \tag{5}$$

Variance is used as a measure of data distribution as this function allows the model to have a clear understanding of the distance of the selected random variables to the mean. This in turn, allows for changes to be made more easily as a higher variance levels are a strong indicator on generating accurate classification models [7].

$$\sigma^2 = \frac{1}{n}(\sum_{i=1}^{n}(x_i - \overline{\mu})^2) \tag{6}$$

The employment of standard deviation allows for its measurement to determine if the mean value is scattered to the values it generates. In addition, a low standard variation acts as a strong indicator which shows a strong classification model as its values are closer to the mean and hence, are more reliable or if changes need to be made in order for the model to be more accurate.

Upon the calculation of all of the segments, the values are also normalised in order to improve the learning process for classification after features have been extracted and selected. This is done through consecutive normalisation of all segments as shown below in (7):

$$f_i = \frac{f_i - f_{min}}{f_{max} - f_{min}} \tag{7}$$

with f_i showing the value of feature one segment whereas f_{min} representing the lowest value of features for all segments and f_{max} representing the largest value of features for all segments.

The relationship and connections within frequency bands of an EEG cap are then placed on a 2D matrix containing the extracted features for all bands. This was chosen due to its structure assisting in being able to recognise different features with more efficiency in comparison to other methods. That matrix is then passed to multi-scale convolutional block of 16, 32, 64 and 128 output channels for finding interconnected correlations and features within the matrix. This is achieved using Pearson correlation to find connection between as seen in Fig 2.2 [18].

Once those features are collected, they are added onto a fully connected layer which

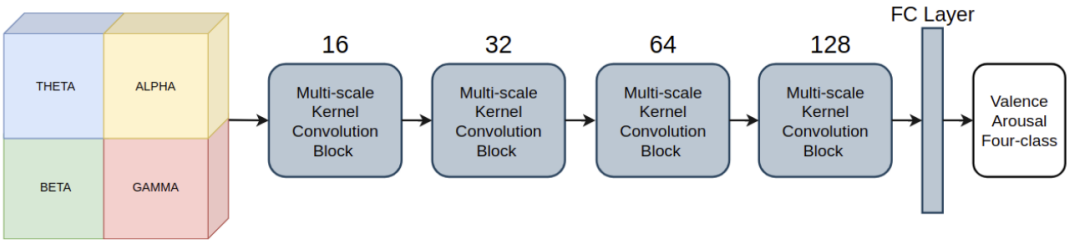

Figure 2.2: Interfrequency relationship correlation method

contains the final feature vector that is classified using softmax for four class valence and arousal classification [6, 20].

Parallel Sequence-Channel Projection CNN

The parallel sequence-channel projection CNN by *Lili Shen et al*, focuses on electrode correlation information through the introduction of the Temporal sub-network comprising of a sequence-projection layer and Spatial sub-network consisting a channel-projection layer which handle the extraction of temporal and spatial information of EEG signals as seen in Fig 2.3 [20]

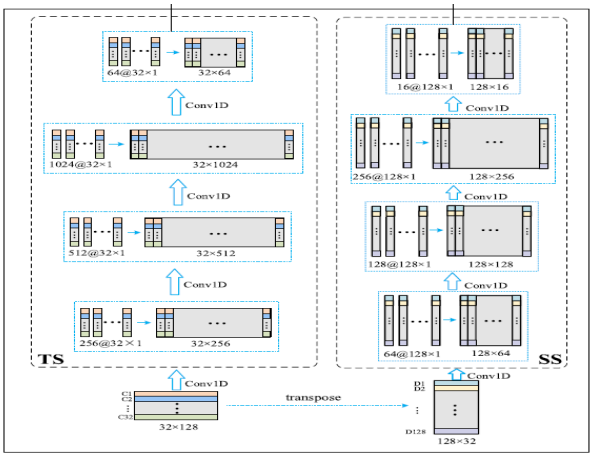

Figure 2.3: Temporal and Spatial Sub Network

This shows the focal point of this experiment is being able to generate a model which is capable of being more accurate than others based on the extraction of high level features based on hidden feature learning from time continuity and space correlation. The sequence-projection layer uses temporal convolutional kernel with the same size as the length of the transmitted EEG sequence to display each sequence individually. After being looked at individually, the shape of the output map is permutated from (32,1,256) to (32,256). This continues with the addition of the 512 temporal convolutional kernels with the sequence of (1,256) and 1024 kernels with the sequence of (1,512) to learn feature temporal representation on a higher level [20]. Once all feature areas are explored, 64 convolutional kernels are implemented within the shape of a (1,1024) sequence in order to reduce the length of outputs for temporal features, resulting in the temporal feature vector which is fed into the fusion classification block.

Spatial Sub Network The spatial sub network consisting of the channel projection layer is used as a form of spatial representation from EEG signals to capture spatial correlation within all channels of the EEG cap. 64 spatial convolutional filters are implemented with a vector of the size of (1,32) to explore each channel in one moment. The stride movement of 1 is still in used to explore each time period individually [20]. Once all time periods are explored and permutation is implemented, similar to the Temporal stream sub network, convolutional filters of higher value are explored to integrate spatial representation as seen from the usage of 128 spatial convolutional filter with the vectors of size (1,64) and 256 spatial convolutional filters with the vectors of size (1,128) [15]. After all areas have been explored, 16 spatial convolutional filters are applied to minimise the length of outputs on the space dimension. Upon analysis from the 4 channel-projection layers, the spatial feature is created and passed onto the fusion classification block.

Fusion Classification Block The classification process introduced and implemented by *Lili Shen et al* uses a fusion classification block to concatenate the vectors to a joint vector as shown in (8) and Fig 2.4 [20]. The fully connected layers then pass on the vector to the softmax classifier (9) that uses fully connected layers to process the fusion classification block to classify the emotional state of the participant. In addition, cross-entropy (10) is used as a loss function and evaluates how well the algorithm goes with the model [5].

$$S - TFV_j = concat[SFV_j, TFV_j] \in \mathbb{R}^{4096} \tag{8}$$

$$y_j = Softmax[FC(S - TFV_j], y_j \in \mathbb{R}^{\nVdash} \tag{9}$$

$$PWTM = argmin(\sum_{j=1}^{n}\sum_{k=1}^{K} -log(p_k)\delta(y_j = l_k) + \alpha||\theta||) \tag{10}$$

with PWTM and θ showing the parameters of a well trained model and the current model. The indicator function δ is used with the mean classified label and a true one amongst the training samples n and class labels k.

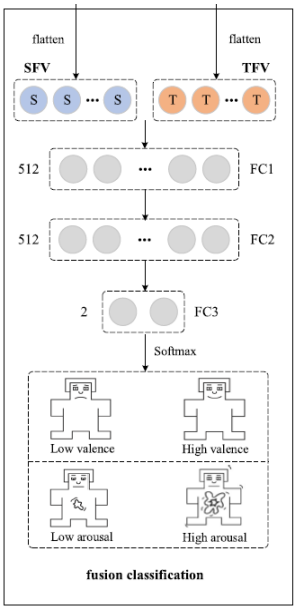

Figure 2.4: Fusion Classification Block

Regional-Asymmetric CNN

Heng Cui et al project uses an RACNN model, as shown in Fig 2.5 [6] and its feature extractors to be able to access features necessary for accurate emotion recognition. Spatial relationships of EEG channels generated during preprocessing were modeled based on the positions of the electrodes within an EEG cap for DEAP datasets ciphered onto a vector of 9 X 9 in order to encapsulate all electrode positions that the EEG cap contains. With DREAMER datasets, 14 electrodes are encoded into a vector with a similar height and weight as the one used for DEAP datasets. From doing so, the channels that are right next to one another maintain their relationships within the matrix and overlapping is prevented with a channel pairing to the left and right hemisphere of the cap is maintained in the vector as well.

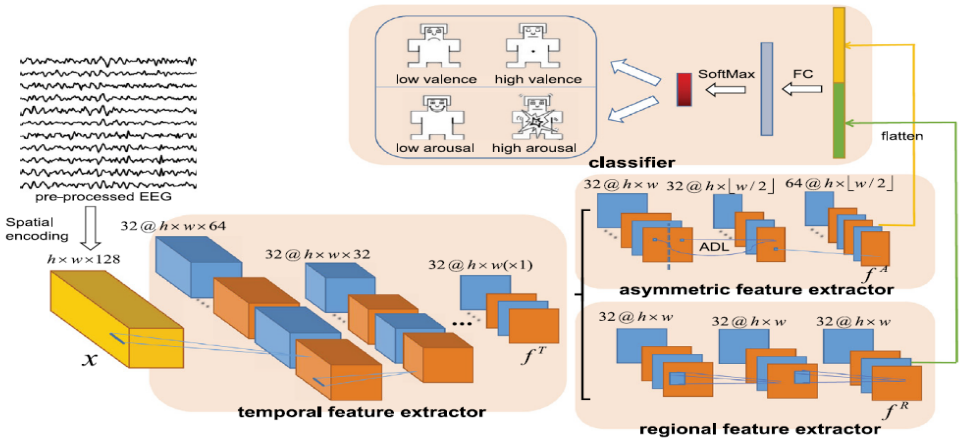

Figure 2.5: RACNN model

Temporal Feature The ability of the RACNN model to be able to learn the temporal features from each channel while also being able to learn higher-level regional and assymetric features from the temporal features captured can allow for the model to be more accurate in emotion recognition while also being able to collect discriminative features [6, 15]. This is achieved from the first major section of the RACNN model which uses input data from each channel with 3D convolutional functions to generate low level feature representations. This is done through the use of the four convolutional layers with 32 temporal kernels with the respective sizes : **1 X 1 X 3, 1 X 1 X 3, 1 X 1 X 5, 1 X 1 X 16** [6]. This in turn, allows for the three smallest kernels to capture high frequency EEG signal representations which contains the most amount of information and the largest kernel to be able to combine them. This breakdown of structure helps short-term and long-term temporal information to be learned more easily. Along with that, the setting of the convolutional step to 2 allows for less calculations and temporal features can be extracted from the input data of the EEG channels. This can be seen from equation (11) below:

$$f^T = E_T(X) = [f_1^T, f_2^T, ..., f_{32}^T] \in \mathbb{R}^{hxw(x1)x32} \tag{11}$$

with f^T representing the temporal feature which is gathered and E_T representing the temporal feature extraction function involving the covolutional layers and $f_i^T \in \mathbb{R}^{hxw}$ shows the temporal feature map of f^T for i.

Regional Feature Extractor The acquisition of the temporal features act as an input for the regional and asymmetric feature extractor of the RACNN model (Fig 2.5) where regional and assymetric features are collected. This is done using two 2D convolutional layers with similar kernel sizes (**3 X 3**: DEAP, **1 X 3**: DREAMER) to understand regional information of the temporal features extracted. In addition, the application of the zero-padding method prevents the loss of edge information to the temporal feature map. This in turn, allows for the regional features to be captured. The implementation used in the regional feature extractor is shown from equation (12):

$$f^R = E_R(f^T) \in \mathbb{R}^{hxwx32} \tag{12}$$

with f^R showing the regional feature and $E_R(\cdot)$ showing regional feature extractor function.

Assymetric Feature Extractor The motivation behind the use of the asymmetric feature extractor is to be able to capture long-term information in the context of EEG signals. This feature extractor was modeled around the idea of being able to distinguish the two different sections of the EEG cap [6]. This was accomplished through the separation of the paired channels based on their positions on the brain scalp on the EEG cap as shown from equation (13):

$$f^A(i, j, k) = f^T(i, j, k) - f^T(i, w + 1 - j, k) \tag{13}$$

where i would cover the entire height section and j would be used to separate the left and right hemisphere which were established at the beginning of the feature extraction

progress, meaning it would cover the boundary areas of $w/2$.

The integration of the separated channels are achieved by using an additional convolutional layer that combines the asymmetric features captured on all of the channel pairs which were generated previously. The number of kernels is set to 64 in order to maintain the level of consistency with the findings on the regional features. This is done by combining the original asymmetric features $\overline{f}^A \in \mathbb{R}^{hx[w/2]x32}$ with the assymetric feature extractor function in (10), resulting in (14):

$$f^A = E_A(f^T) \in \mathbb{R}^{hx[w/2]x64} \tag{14}$$

Heng Cui et al explores this in Fig 2.6 by using findings from the asymmetric feature extractor and the regional feature extractor as the flattened vectors which are added into the final feature vector as shown in (15) and connected into one layer. That layer is passed onto a softmax classifier [6, 17, 18] (16) which determines the probability of a feature belonging to the classes of low valence, low arousal, high valence or high arousal.

$$o = FC(f^R||f^A) \in \mathbb{R}^{\text{illegible}} \tag{15}$$

$$\begin{aligned} y &= Wo^T + b = [y1, y2, ..., y_c]^T \in \mathbb{R}^{\text{illegible}} \\ P(c|x) &= \frac{exp(y_c)}{\sum_1^C exp(y_i}, c = 1, 2, ..., C \\ \hat{e} &= argmax P(c|x) \end{aligned} \tag{16}$$

Where the findings from the Assymetric Feature extractor and Regional feature are concatenated. Once calculated, linear transformation is used to preserve the vector shape. This is calculated with weight matrix W and b.

The value of y is used in respect to the number of emotional categories which is set to 2 due to the 2 class valence and arousal classification to perform softmax.

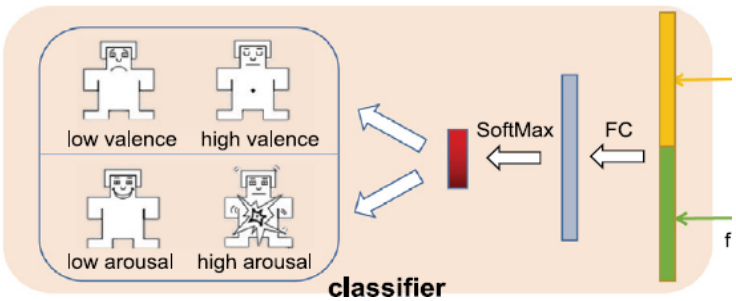

Figure 2.6: RACNN model classifier

With a focus on face data along with EEG signals, the feature extraction methodology implemented by *Elham S. Salama et al* with their 3D CNN model targets to integrate EEG signals along with face signals to achieve emotion classification as this allows for more accurate results to be generated when classifying an emotion for the participant when they are exposed to a particular stimuli as shown in Fig 2.7 [19]. The initial process to achieve this involves achieving input fusion which combines data from multiple sensors, which in this context involves the face chunks along with the EEG chunks that were captured using the EEG cap.

The initial stages involved with the feature extraction process involve the combination of the input EEG and face chunks. Fusion chunks are generated by combining these chunks on every 5 second interval from the main trial with the first 3 seconds being removed due to them not including any important data. The chunks are made by joining each chunk above each other. Similar to [6, 8, 15, 18], this project also uses the DEAP dataset framework as a means to generate the chunks needed to input into the main network for classification with the creation of 7680 samples, based on the frame rate which was agreed upon during pre-processing. This allows for 1536 samples being divided on each frame and the entire fusion input being divided into 12 equal chunks.

The final section of feature extraction involves the culmination of trained input data being used for training the main 3D CNN model for emotion classification and the implementation of . This involves the initial weights being replaced by weights of a trained model. As a result, this allows the main model to be able to make classifications without needing time for it to be trained. This logic is applied to the fusion system with the transfer of the trained data from the respective 3D CNN systems for the original face data and EEG signals, containing their emotion classification values which are then fed into the main network alongside the original fusion chunks which were trained within the main network through the conventional 3D CNN. This method is applied to reduce generalisation error and improve the overall network accuracy. Once classifications are generated for the original values, score fusion methods of stacking and bagging are used to further enhance the final classification accuracy of the model.

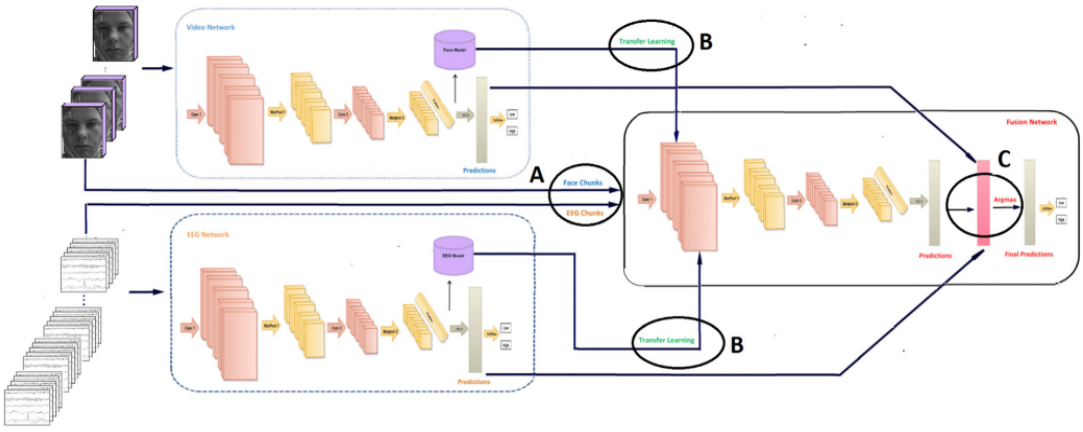

Figure 2.7: 3D CNN framework

Stacking

The learning technique of stacking is particularly used on the main 3D CNN to learn from previously trained models and is able to use that information to generate the value from those models that has the strongest value. This is achieved through the use of the Maximum A Posteriori(MAP) rule as seen in Fig 2.8 [19]. Methods for stacking involve the use of model averaging ensemble, weighed sum ensemble and grid search ensemble. This is particularly useful when looking at 3 sets of outputs which are all generated the training of data as done with this project [19]. The final combined classification value for the model through stacking is generated by finding the max value from the classification values for valence and arousal for the EEG and facial 3D CNN systems alongside the fusion chunk systems, as shown in (17):

$$P = arg\,max(S(i), F(i), E(i)) \tag{17}$$

with $S(\bullet), F(\bullet) and E(\bullet)$ representing the classification values of 3D CNN systems of the fusion chunks, face data and EEG signals respectively for valence and arousal with P representing the final classification value generated from those 3 systems.

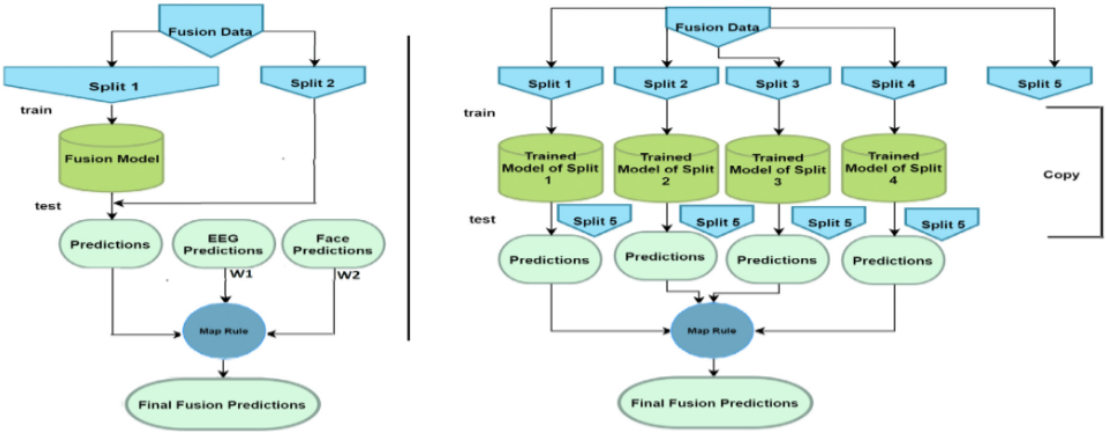

Figure 2.8: Stacking and Bagging Models

Bagging

The main concept around bagging involves the creation of several datasets from the split of the fusion chunk data originally fed into the network with a separate classifier assigned for each split. This is achieved using k-fold-cross-validation with k being set to 5. One of the sections that is split is used as testing data for the training data. Each of the splits generated apart from the one left out for testing is trained and tested using the testing data, resulting in 4 separate classification values. MAP rule is used to pick out the best classification value that is generated from the comparison of the classification values generated from all of the data that was split.

Elham S. Salama et al uses argmax classifier between the classification values of the integrated model, the face chunk model and the EEG signal model whose value is then taken in by the soft-max classifier to classify an emotion based on high or low valence and arousal. The changes with the classification process, however starts with the implementation of different types of models based on the feature extraction methods with different

stacking methods resulting in different accuracy levels for emotion recognition as shown in Fig 2.7.

2.3 Recurrent Neural Networks

2.3.1 Pre-Processing

Heng Cui et al display this through an emotion recognition model using a GRU and minimum class confusion (MCC). Data pre-processing for this model involves the use of SEED dataset and MPED dataset to capture and process information with hand-crafted features before being fed into the model for further feature extraction. For the SEED dataset, 15 participants are selected and their data recorded using 15 movie scenes in 3 different sessions. 5 clips are of positive, neutral and negative tones equally and each scene lasts 5 minutes [5]. Each participant therefore has 45 segments. EEG signals are recorded with 62 channels with a sampling rate of 1000 Hz which is downsized to 200 Hz. Samples of data congested with Electromyogram (EMG) and Electrooculography (EOG) are discarded. In addition, the EEG data was broken down into 1 second data samples without any overlap. [5]

With the MPED dataset, EEG signals of 23 subjects were recorded while watching 28 movie scenes, with 4 of them each conveying one of 7 types of emotion : joy, funny, neutral, sad, fear, disgust and anger. Signals were also recorded with 62 channels with a sampling rate of 1000 Hz. Independent component analysis (ICA) was used to remove EOG remnants and the main signal was broken down to 1 second signals.

DE features extracted for each frequency band of EEG: delta, theta, alpha, beta and gamma was used as the hand crafted features that are implemented before being fed to the network for the SEED dataset.

For MPED, however, High Order Crossings (HOC) features, which uses a time series to find oscillating patterns within the input it gets. This is seen in equations (18) and (19):

where (gradient) acts as a difference operator which is used to find a relation within all of the points that is being inputted to generate HOC features.

Tong Zhang et al on the other hand, implements a different approach towards pre-processing with a stronger focus on the spatial dependencies and temporal variations in an EEG signal with a spatial-temporal recurrent neural network for emotion recognition [15, 20]. The model proposed here is tested on the SEED dataset across 15 participants in 2 time sessions. EEG signals are gathered using emotional movie clips to participants. Due to the possibility of data that is captured through EEG signals can be contaminated, the usage of 4 directional RNNs are introduced which go through a spatial region at a time from 4 angles. This allows to reduce the effect of noises while also helping to generate a relationship within each sector of the four directional RNNs [23].

Huiping Jiang et al uses two different variants of LTSM recurrent neural network with a focus on wavelet transform for preliminary analysis before training the model for feature extraction. The experimentation framework begins with 12 videos being selected with 6 of the 12 videos were chosen from movies. Initial emotional feelings of participants were

recorded by themselves using the 9 point SAM scale [2,12]. Upon further analysis, 6 videos were selected to be shown to the participants. This was mainly done so the videos can be kept as short as possible to avoid having multiple emotion readings on one signal [12]. Each scene selected was 3 minutes long with a blank screen shown to users for the starting 10 seconds and then users gave their feedback within 30 seconds after watching the video.

The signals were recorded using the Synamps2 amplifier and Scan4.5 software, a cap containing 62 electrodes with 1 computer used to gather EEG signals [12]. Signals that were collected were acquired using the Neuralscan-64 system. The sampling frequency was set to 1000 Hz. Electrode distribution for the experiment followed the widely popular 10/20 system electrode placement method. Digital filtering was implemented as a final step before feature extraction takes place.

Anumit Garg et al propose a merged LSTM model which handles the work of each channel separately and is merged together after feature extraction for generating the final result. Data of 32 participants is collected using the DEAP dataset [6, 7, 18] and preprocessed with MATLAB and some Numpy formats from Python. Due to the limited hardware resources for this project, the usage of processes to remove unwanted noise and data is done using dimensionality reduction. [7]

2.3.2 Feature Extractor and Classification

Feature Extraction within RNN centres around the use of sequential data and finding relationships within that sequence in order to generate a result. The only downside with its implementation is that RNN are known to have a massive reduction in their gradient value during backpropagation, which leads to the network not being able to learn effectively from the data it is provided.[,] This same problem is mentioned and acted upon with several experiments that look at the implementation of different variations of RNN which look to store required information in order for the network to only learn with features required to make an accurate classification from the model.

GRU and MCC Emotion Recognition Model

Heng Cui et al show this by modeling their feature extraction methods for their model to cater to short-distance and long-distance dependence between channels using GRU and MCC. Firstly, electrodes are reordered to the 10-20 system [1, 5, 6]. The implementation of GRU is done during the extractor phase with the implementation of the reset gate and the update gate. Reset gate is used to learn the spatial information that is present with each of the input channels have with one another, allowing for important features to be captured. Reset gate uses the activation function of sigmoid on the weighed sum of the EEG channel with the activation of the channel before as input [15]. The function can be seen in (20):

$$h_i = \sigma(w_h \cdot x_i + w_h \cdot h_{i-1}) \tag{20}$$

Once the last activation is completed, the activation sequence of h is broken down to a feature vector *h'*. Once that is completed, a fully connected layer with a rectified linear unit(ReLU) is used to reduce feature dimension and acquire the final representation value which is sent to the classifier [5].

There are two forms of classification applied to the model here as seen from Fig 2.9

which gives a total breakdown of the proposed model. The first process is the use of a simple linear transform as the classifier. The given output is passed to a softmax [6, 19] layer which is expressed in (21):

$$P(j|x) = \frac{exp(z_j)}{\sum_{i=1}^{R}}, j = 1, 2, ..., R \qquad (21)$$

where the probability of x belonging to a class j is determined and gives the emotion of a user based on the signal provided.

The cross entropy loss [5, 20] function is calculated with the ground truth labels to calculate the errors within the network and evaluate the performance of the model in generating a correct classification value in the softmax layer. This allows for the emotion classification for source samples are efficient and the learned depth features are related to emotion. The cross-entropy can be seen in (22) and (23):

$$L_{CE}(X^S|\theta) = \sum_{i=1}^{B_S}\sum_{j=1}^{R} -r(l_i, j)log\mathbb{P}(j|X_i^s) \qquad (22)$$

$$r(l_i, j) = \begin{cases} 1 & j = l_i \\ 0 & otherwise \end{cases} \qquad (23)$$

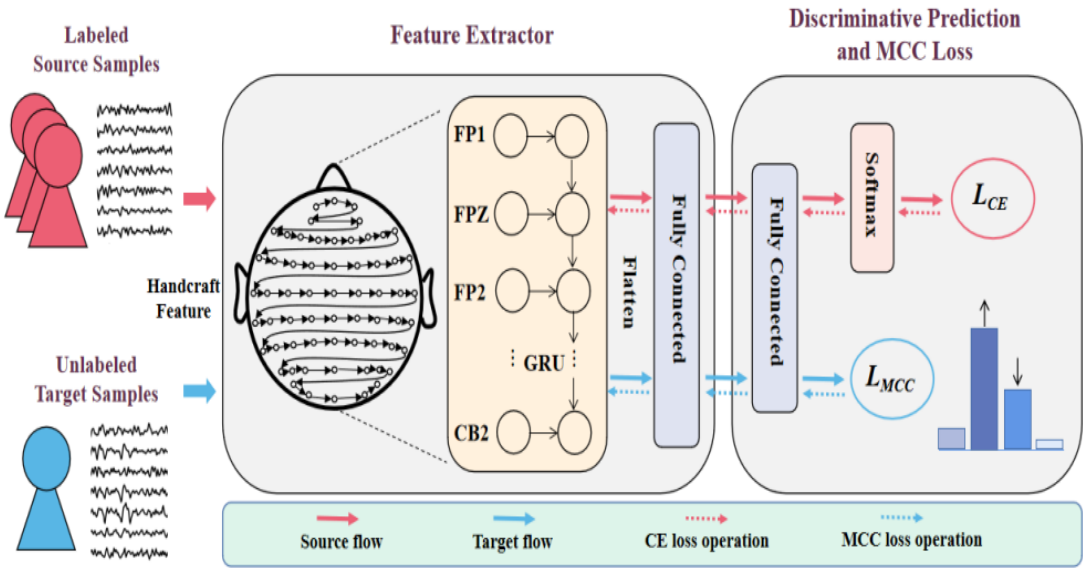

Figure 2.9: GRU,MCC Emotion Recognition Model

The classification process behind MCC loss is determined in order to be able to improve the performance of the subject independent function by reducing class confusion by using Probability Rescaling, Class Correlation, Uncertainty Reweighing, Category Normalisation and finally, MCC(24) [5]:

$$L_{MCC}(X^t|\theta) = \frac{1}{R}\sum_{j=1}^{R}\sum_{j\neq 1}^{R}|\tilde{C}_j j'| \qquad (24)$$

Spatial-Temporal RNN

The Spatial-Temporal RNN as seen in Fig 2.10 by *Tony Zhang et al* employs a different application of feature extraction methods used to specifically outline the similar aim to *Heng Cui et al* by trying to learn spatial dependencies and temporal dependencies and model their relationship within an EEG signal reading. The use of DE is heavily used in the feature extraction process of the model with the DE values of the EEG signals which are calculated based on the five frequency bands of an EEG signal: Delta, Theta, Alpha, Beta and Gamma [10, 23]. For specifically longer pieces of EEG sequence, a Short Time Fourier transform function is used with a nonoverlapped Hanning window of 1 second is used to extract the frequency bands and DE is collected for each frequency band. Discrete sequences which correspond to each time period based on the Hz values of the frequency bands are generated [23]. A sliding window function of 9 seconds is implemented with one step to find temporal dependencies that are involved with finding human emotion at a specific time [18].

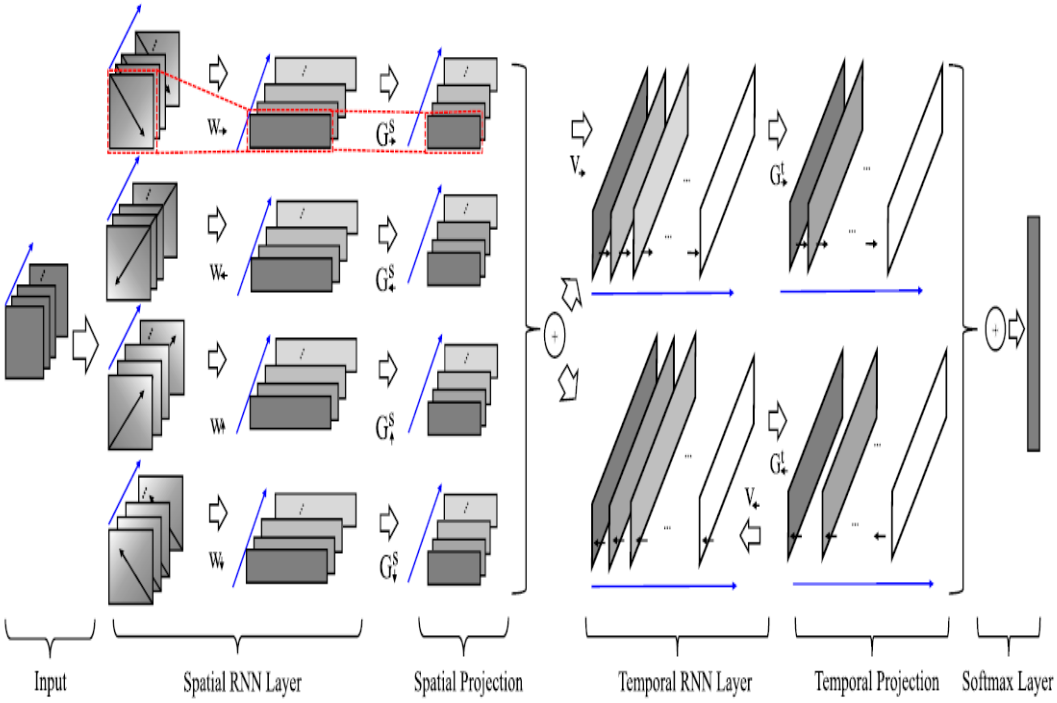

Figure 2.10: Spatial-Temporal RNN

Spatial dependencies are generated with a graph that is used to show the spatial elements till the end of the graph, which is defined through X_t. The traversal through the graph is used to find hidden states within the graph that contain features for emotion representation. This can seen with function (25) and (26):

$$h^r_{tij} = \sigma_1(U^r\ x_{tij} + \sum_{k=1}^{h}\sum_{l=1}^{w} W^r h^r_{tkltij,tkl} + b^r) \tag{25}$$

$$e_{tij,tkl} = \begin{cases} 1 & if(k,l) \in N^r_{ij} \\ 0 & otherwise \end{cases} \tag{26}$$

where x_{tij} and h^r_{tij} are used as a representation of the input and hidden node within the spectrum of the tth slice. N^r_{ij} is used to represent the points before i and j within the traversing direction of r. U^r, W^r, b^r are used to show the learneable parameters in Spatial Recurrent Neural Network Layer(SRNN) and $\sigma_1(\cdot)$ show the nonlinear function for hidden layers within SRNN [15, 23]. Therefore, h^r_{tij} collects spatial information through backpropagation in regard to the previous elements before it. Once those hidden states are found, the feature extraction methods of DE are applied to find the emotion representation features within the hidden states that are discovered through the use of projection matrices.

The projection matrices that are generated are used to access the hidden states of emotion representation on each traversing region [23]. This is done by changing the hidden states output to correspond to h x w for one traversing direction. The concatenated feature vector generated from one particular traversed direction is found in (27):

$$s^r_{tl} = \sum_{i=1}^{K} G^r_{il} h^r_{il}, l = 1, ..., K_d \tag{27}$$

where G^r_{il} is used as an indicator that shows the number of hidden states on a particular traversing direction after projection and s^r_t is used to highlight the final feature vector for one traversed direction.

Emotion Computing Model using LSTM

The various implementation of the LSTM model have been used to further refine the process of extracting features from the ground truth. This is evident from the application of the Wavelet Transform (WT) LTSM and the ALL LSTM with both models looking to implement different feature extraction methods to better classify emotions generated from EEG signals.

ALL LSTM As seen in Fig 2.11 [12], the implementation of the ALL-LTSM model looks to use the preprocessed EEG signals that are given at the beginning of the model. The information is then passed onto the LSTM layer which is responsible for extracting contextual related features which include time-domain features like mean, variance and standard deviation [18]. Once those features are extracted, similar to the standard LSTM network, the fully connected layer is responsible for combining the features learned from each LSTM unit which correspond to different areas of the preprocessed EEG signal. Once completed, the process of the classification of emotions based on the output given

is determined.

The classification process implemented in the ALL LSTM model involves the use of the SoftMax classification as shown in (21) [5]. This model is used to determine the emotional category of the output that has been given based on the features extracted on the preprocessing signal.

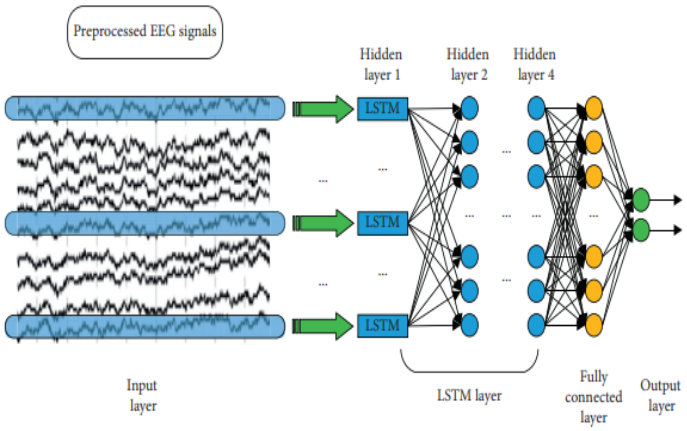

Figure 2.11: All LSTM model

WT LSTM The WT LSTM, on the other hand, uses different forms of feature extraction to further refine the model it has. This can be shown through Fig 2.12 [12] where the process of emotion recognition based on WT is performed first and then used on the LSTM classification model. This was mostly done through the use of WT which breaks down the input preprocessed EEG signals to small range frequency bands of Delta, Theta, Alpha, Beta and Gamma. Once that is finished, the process of wavelet coefficients of each layer of the frequency band generated previously and feature extraction methods of wavelet transform are implemented using the same parameters. This includes band energy(E)(28), band energy ratio(REE)(29), logarithm of the band energy ratio(LREE)(30) and the DE(1):

$$E_i = \sum_{j=1}^{n_i} d_{ij}^2 \qquad (28)$$

where the Energy of band i is determined using the sum of the continuous wavelet coefficients of j in the layer i till n_i which is used to display the number of coefficients that were generated in the layer i

$$REE_i = \frac{E_i}{\sum_{j=1}^{n} E_j} \qquad (29)$$

where the band energy ratio of band i is found from the standard division of the energy value of band i from the sum of the energy value of the rest of the bands.

$$LREE_i = log_{10} REE_i \qquad (30)$$

where the logarithm of the energy ratio of band i is calculated based on the logarithm value of 10.

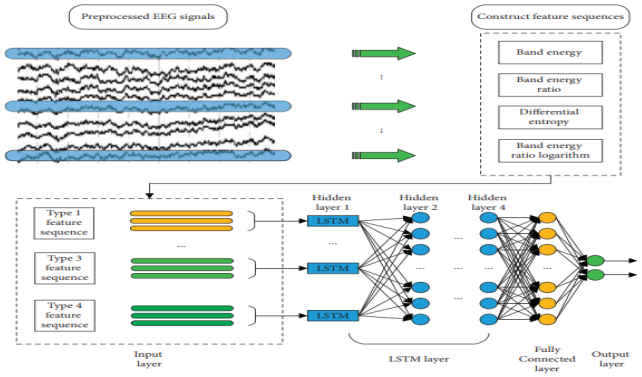

Figure 2.12: WT LSTM model

The DE is found based on the distribution of the signal where Gaussian distribution is used as the function as the input for DE if the signal has a different form of distribution.

The wavelet features [7] are then passed on to the LTSM layer, where similar to the ALL-LSTM model, gather context-related features from the sequences of the wavelet features. The fully connected layer connects the features extracted from each sequence and converts them the desired output.

The classification process for the WT LSTM model is generated from the output of the extracted features being combined together in the fully connected layer and being classified to an emotion using Softmax (21).

Merged LSTM Model

The feature extraction and classification framework of the Merged LSTM model as seen in Fig 2.13 by *Anumit Garg et al* describe the implementation of DWT and dimensionality reduction as the main feature extraction process.

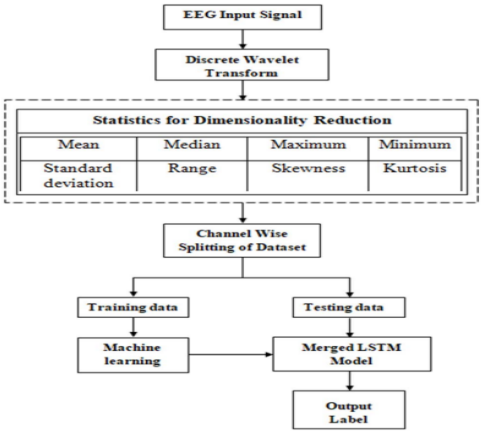

Figure 2.13: Merged LSTM Model Network Architecture

DWT is used to break down the signal through subsets and to remove noise from the preprocessed EEG signal. In addition, this allows time series features to be collected from the EEG signals [7]. The process of feature extraction is further extended through statistical features like mean and standard deviation used for dimensionality reduction in the EEG signals that have been extracted through DWT features.

The classification process involved with the combined LSTM model splits the extracted features into training and testing datasets. Once that is done, the 40 channels that have been preprocessed are fed into the merged LSTM model as shown in Fig 2.14 with each channel being individually processed by an LSTM model in order to learn more about each of the network in correspondence to the channel it is learning from. In addition, this process allows the model to capture each channel's responsibility in helping to classify the emotional state of a participant with more efficiency. Once the channels are merged, 2 fully connected layers are used to generate a feature vector which is then given as the final output for the Soft Max classifier [6, 19] (16).

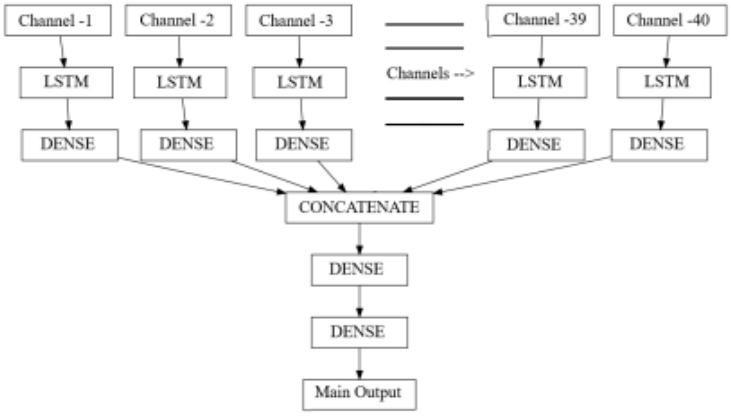

Figure 2.14: Merged LSTM model

2.4 Self Attention Models

Self attention Models is another state-of-the-art artificial neural network where its application focuses on language understanding. This is mainly done through its ability to find connections within different sections of input by allowing them to highlight the sections of input sequence that contain the most amount of information. Its recent implementation on image processing alongside other state-of-the art models shows its implementation is capable of performing just as well in capturing information from EEG signals and its different mechanisms during preprocessing and feature extraction can lead to a model which is very accurate and easy to maintain in the future [11].

2.4.1 Preprocessing

Ziyu Jia et al explore the implementation of self-attention mechanisms through their GraphNet Spatial Temporal Graph Convolution Network (GCN) with graph and temporal convolution and a spatial-temporal attention mechanism to find relationships and correlations between sleep stages and EEG signals.

The preprocessing steps include the collection of the data from the Montreal Archive of Sleep Studies (MASS) dataset which consist of Polysomnography(PSG) readings from 62 participants with every recording containing 20 EEG, 2 EOG, 3 EMG and 1 ECG channel. Its initial processing steps using bandpass filters of 0.30 - 100 , 0.10-100 and 10-100 Hz for EEG, EOG, ECG and EMG signals respectively [11]. This in turn, allows for signals that cater towards EEG to be identified and used more easily for feature extraction with DE features [5, 11].

Dongdong Li et al use the public emotion datasets of DEAP and DREAMER for input data for their spatial-frequency convolutional self- attention network(SFCSAN) model. The preprocessing of the EEG signals being processed for each participant involve the sampling frequency of the EEG signals to 128 Hz with EEG signals from 4.0-45 Hz are gathered EOG signals are removed. In addition, training sets and models for ocular features DEAP and DREAMER are removed. Butterworth filter [18] is then used on the captured EEG signals to ensure separation to the bands of theta, beta, alpha and gamma. Non-overlapped Hanning window of 3 seconds was used to separated the EEG signals of each frequency band.

Wei Tao et al use DEAP and DREAMER datasets [6, 19] for their channel-wise attention and self-attention model(ACRNN) for emotion recognition using EEG signals.

The preliminary steps before preprocessing involved in this experiment include retrieving data from 32 participants watching 40 music videos. EEG signals are of 32 channels and fully focus on emotion recognition. The signals are downsampled from 512 Hz to 128 Hz and baseline signals are removed as a means to improve EEG emotion recognition [21]. This is achieved by using the average baseline signal for the duration of the trial through (31):

$$\overline{X}_B = \frac{\sum_{i=1}^{T_2} X_i}{T_2} \tag{31}$$

with $X_B \in \mathbb{R}^{MXL}$ is representative of the baseline signals throughout T_2 with the signals at a particular time being found using X_i.

Once the average values are found for baseline signals, they are removed from the trial EEG signals by its segmentation into small slices to identify the area where the average baseline signal is located and to remove them [21]. The improved value is then sent to ACRNN for feature extraction and classification. This is seen from the representation of $X_j(j = 1, 2, ..., T_3)$ and its use in (32).

$$X'_j = X_j - \overline{X}_B \tag{32}$$

2.4.2 Feature Extractor and Classification

GraphNet GCN

The feature extraction process involved with GraphNet that is used by *Ziya Jia et al* consist of the collection of DE features for each channel as shown in Fig 2.15 .Features for each channel are collected on the crossed frequency bands of 0.5-4, 2-6, 4-8, 6-11, 8-14, 11-22, 14-31, 22-40 and 31-50 Hz [11]. The first important section involved with the feature extraction is the understanding of the relationship between the channels and the feature matrix [11, 21]. The values generated through this are added onto the adjacency matrix which is an undirected graph $G = (V, E, A)$ where each separation within the matrix highlights 30 seconds of the main EEG sequence. This is generated from the function as shown in (33):

$$A_{mn} = g(x_m, xn) = \frac{exp(ReLU(w^T|x_m - xn))}{\sum_{n=1}^{N} exp(ReLU(w^T|x_m - xn))} \tag{33}$$

where the connection between each node of the feature matrix with one another is achieved using the layer neural network containing a learnable weight vector which corresponds to each of the nodes that are being compared with one another. A ReLu activation function is used that takes in the softmax value of the weight vector and the distance between each of the nodes is generated and the row of the matrix is normalised [5, 11]. This ensures that the A_{mn} is ≥ 0 and the adjacency matrix only contains positive values.

The minimisation of the loss function is used to update the weight vector as shown in (31), which ensures that the smaller the distance is between the nodes, the larger A_{mn} is and vice versa. This allows for the scatterness of the adjacency matrix to be controlled as A_{mn} is directly used alongside λ, a regularised parameter that is ≥ 0 [11].

$$\lambda_{graph-learning} = ||x_m - xn||_2^2 A_{mn} + \lambda ||A||_F^2 \tag{34}$$

Spatial Temporal Graph Convolution Since the adjacency matrix in this model is a learnable matrix, an identification of a sleep stage automatically leads to an adjacency matrix being fed to the spatial temporal graph convolution network [6, 15, 20]. It is then implemented for the second important feature extraction action of this model, the extraction of spatial and temporal features through graph convolution. This is done through the use of the Chebyshev expansion of the Laplacian matrix which takes in the adjacency matrix in this project in the convolution kernel as shown in (32). This allows the convolution kernel to identify information of the neighbors of each node is

extracted [11].

$$g_\theta *_G x = g_\theta(\mathbf{L})x = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{L})x \tag{35}$$

where for the context of this project the sleep stages extracted from the adjacency matrix, $\tilde{X}^{l-1} = (\tilde{X}_1, \tilde{X}_2, ..., \tilde{X}_{Tl-1} \in \mathbb{R}^{NXC_{l-1}XT_{l-1}}$ are used with C_{l-1} correspond to each channel for each node and T_{l-1} is used to define the temporal area for the sleep stage. The channel filter C_l is used and the information for 0 - K-1 nodes is found.

A standard 2D convolution is applied to extract temporal information on the basis of finding enough spatial features that have been extracted from each sleep stage network [11, 18]. This can be shown from the formula (36) that highlights the temporal convolution function that takes in the spatial convolution used for a particular stage.

$$X^(l)) = ReLU(\Phi * (ReLU(g_\theta *_G \tilde{X}^{(l-1)}))) \in \mathbb{R}^{NXC_lXT_l} \tag{36}$$

This is done with the use of the ReLu activation function with Φ used to define the convolution kernel's parameters which are the output of the convolution operation that extracts spatial features as seen from $*$.

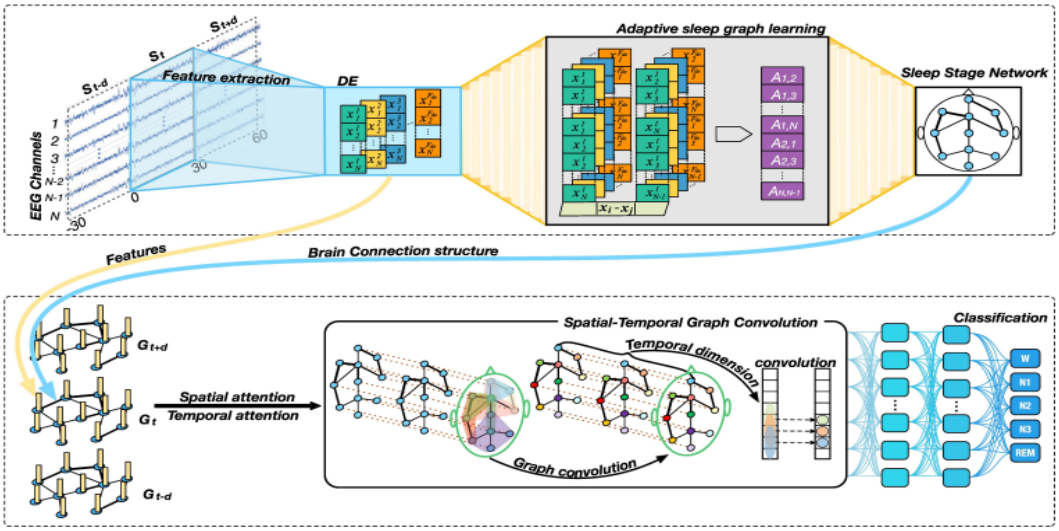

Figure 2.15: GraphSleepNet framework

Spatial-Temporal Attention The final feature extraction phase of the GCN involve a spatial temporal attention model which are used to capture attentive spatial dynamics as shown in (37) and (38):

$$P = V_p \cdot \sigma((X^{l-1}Z_1)Z_2(Z_3X^{l-1})^T + b_p) \tag{37}$$

$$P'_{m,n} = softmax(P_{m,n}) \tag{38}$$

where the sigmoid activation function [9] in (37) is implemented as a means to generate the correlation between nodes m and n through the learnable parameters of the feature

matrix, the channels and the temporal areas being implemented with the temporal sleep state vector, allowing for the different spatial motions to be captured and (38) to be used alongside the newly generated attention matrix to be normalised based on the new changes.

The temporal attention sector,on the other hand, focuses on finding dynamic temporal information within the sleep stage networks. This is done using similar mechanisms as the spatial attention model where the learnable parameters however focus on the temporal dimensions and the changes within the channels with $Q_{m,n}$ generated from (39) allows for the normalisation of the temporal attention matrix of Q with changing situations in (40) [11].

$$Q = V_q \cdot \sigma(((X^{l-1})^T M_1)M_2(M_3 X^{l-1}) + b_q) \tag{39}$$

$$Q'_{m,n} = softmax(Q_{m,n}) \tag{40}$$

Once calculated, these changes are the final input that is pushed onto the fully connected layer and the softmax classifier [17, 18] to determine the sleep stage for each participant.

SFCSAN

Dongdong Li et al proposed a different structure in the capture of EEG signals, however, with a focus on specific extraction of DE features (1) that served for each of the frequency bands the main EEG signal was broken down to during preprocessing for emotion classification [15]. The extraction of the features involves the use of PDE which is applied for learning uninterrupted information. For instances of a certain length of EEG signals that uses gaussian distribution, (41) is used with $f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ and DE eventuating to:

$$\begin{aligned} DE &= -\int_{-\infty}^{+\infty} (\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}) log(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}) dx \\ &= \frac{1}{2} log 2\pi e\sigma^2 \end{aligned} \tag{41}$$

The standard deviation of the DE features from the trial data with the baseline DE features from baseline before trial are collected as final classification features. The obtained DE features are then fed to the Z-score normalisation method which improves the classification model's connectivity speed with the other vectors for each frequency band through the removal of elements that disrupted the EEG signals upon the use of the DE feature extraction method [15]. As a result, four vectors $X^\theta, X^\alpha, X^\beta, X^\gamma$ are generated as shown in Fig 2.16 with each vector representing the feature vectors for each frequency during a video that is used to gather and process EEG signals. They contain the features that correspond to each band on a certain time period and their dimensions. These vectors are then passed on to the main model which is used to find spatial relationships corresponding to each frequency band vector to the Parallel Convolution Neural Network(PCNN).

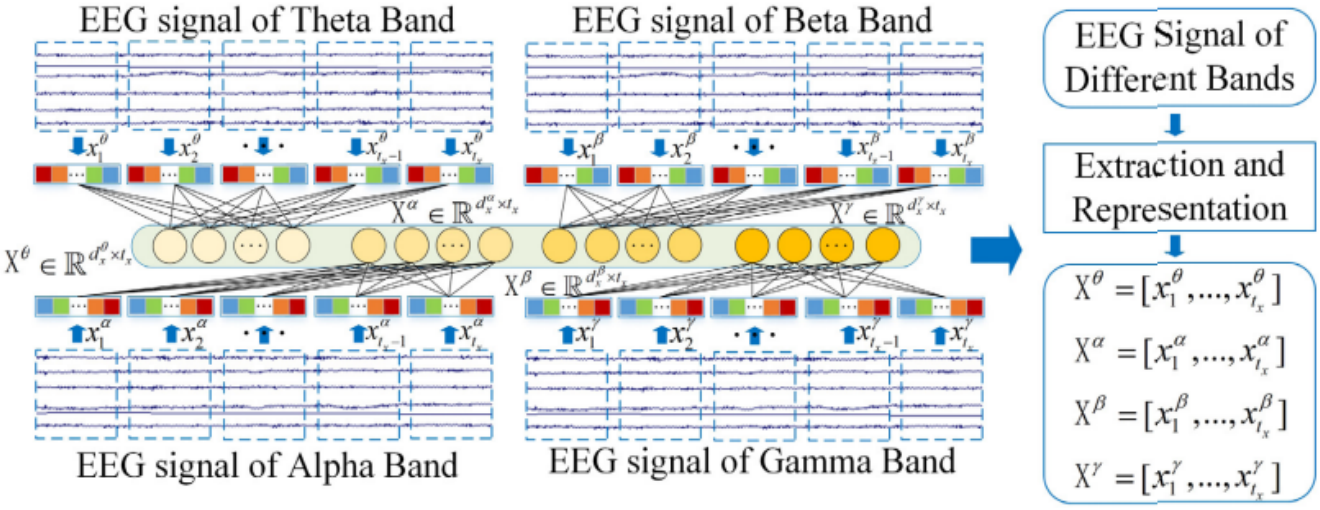

Figure 2.16: SFSCAN Feature Extraction framework

PCNN The feature vector for each frequency is then assigned one PCNN layer which then used to further traverse through it and find further information within itself. This is done using convolutional filtering amongst the 3 PCNN layers with the first layer consisting of 64 kernels which are then doubled across every layer that follows [15]. Scaled experimental Linear Units (SELU) are then used as a method to reduce the likelihood of exploding and/or vanishing gradient as the sizes of kernels grow larger and feature vector for each frequency is kept intact in order for them to integrate with one another [7, 15]. The feature vectors of all frequency for each video are further enhanced to be understood on a conceptual level when passed to the intra-frequency band self-attention model to combine the findings for each frequency band matrix to give a final indication of the information specific to that frequency band.

Intra-Frequency band self-attention The intra-frequency band are used as a catalyst to develop a self-attention map to investigate cross-channel information and to provide its unification. This is done from the discriminative features found [6, 15] in the previous PCNN layer using 1 X 1 convolutional kernels being applied into 2 feature spaces, f_b, g_b, respectively. These feature spaces are then used to find cross channel information through the use of matrix multiplication with higher values allowing the model to find areas of the feature space that contain the most information. In order to reduce computational cost for classification, the number of channels used during the concatenation of the 2 feature spaces is reduced [15]. The softmax classifier (39) is applied to generate the importance of each feature within each matrix based on their relationship with other features.

$$\beta^b_{j,i} = \frac{exp(f(U^b_i)^T g(U^b_j))}{\sum_i exp(f(U^b_i)^T g(U^b_j))} \tag{42}$$

where *iandj* highlight the electrode location and the region of the feature space for a frequency band and $\beta^b_{j,i}$ highlight the importance weight for a particular electrode and its region when going over the frequency band, b.

Once the importance weights are calculated and the attention represented for each frequency band, the discriminative features from the previous PCNN layer for each frequency band are passed onto another feature space to perform matrix multiplication to generate a self-attention map for each frequency band which is then passed for the final section of feature extraction through inter-frequency band mapping.

Inter-frequency band mapping The inter-frequency band mapping model as shown in Fig 2.17 applies the learning of different EEG feature representations from different regions, allowing it to better represent features on a wider scale [11, 15, 19]. This is evident from its application on the SFCSAN which flattens the self-attention map from all frequency bands and concatenates them into a fully-connected layer that maps all features into one vector for classification as seen in (43):

$$Output(f, g, h) = Concat(band^{\theta}, band^{\alpha}, band^{\beta}, band^{\gamma})W \tag{43}$$

where the flattened bands are represented with$band$ and W represents the weight matrix of inter-frequency band mapping [15].

The classification of the EEG signals is determined as follows:

$$p(y'|S) = softmax(W_O O + b_O) \tag{44}$$

$$y = argmax_{y'} p(y'|S) \tag{45}$$

where the learnable parameters of the output layerW_0, b_0 are used alongside the final vector $O = Dropout(Output(f, g, h))$ for classification with a dropout probability of 0.5 for the nodes in the vector [15].

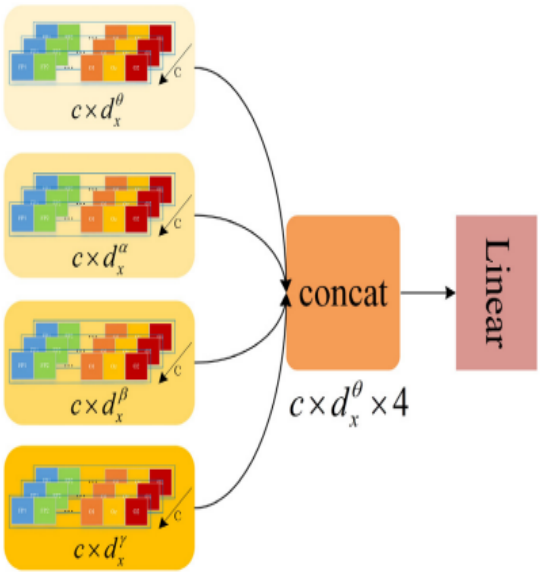

Figure 2.17: Inter-Frequency Mapping

ACRNN

Spatial Feature Extractor The feature extraction of this framework employs channel-wise attention, CNN, RNN and a self-attention mechanism to extract spatial features and then temporal features as seen in Fig 2.18.Channel-wise attention is used to identify channels that are more important from the temporal divisions that were made during preprocessing. This is done from the use of mean pooling to each channel of the collected EEG samples $s_j(j = 1, 2, ..., m)$ with m corresponding to the number of channels of sample s to collect channel-wise statistics in (46):

$$s^- = [s_1^-, s_2^-, , , , s_m^-] \tag{46}$$

Two fully-connected layers are used to improve generalisation through the gating mechanism of the channel-wise attention model to determine the importance of each channel for the EEG sample through the probability distribution set [21]:

$$\upsilon = softmax(W_2 \cdot (tanh(W_1 \cdot s^- + b_1) + b_2)) \tag{47}$$

where the expression $tanh(W_1 \cdot s^- + b_1) + b_2)$ encapsulate a dimensionality reduction layer which allows the probability distribution set υ to be generated based on the mean of a channel in an EEG sample with the dimensionality increasing layer, comprising of the parameter W_2 and bias b_2

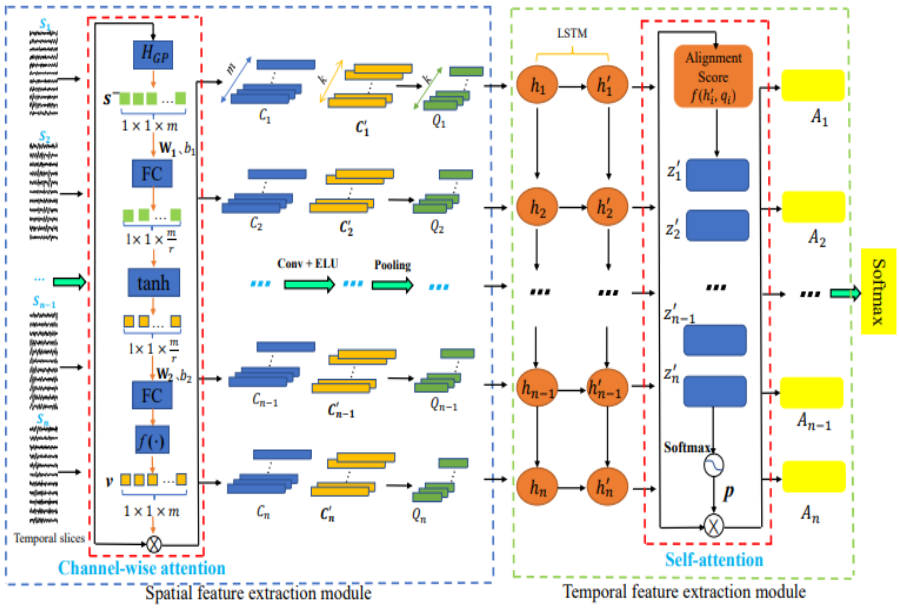

Figure 2.18: ACRNN model

The newly generated set is then used to extract attentive channel features for each channel through:

$$c_j = \upsilon_j \cdot s_j \tag{48}$$

where C represents the features extracted for each channel c_j through the multiplication of each channel of the sample and υ [11].

CNN is then used to explore and collect spatial information based on the channels identified for importance. Convolutional kernels with the same height as the number of channels are made with enough amount of width to capture temporal data as well [21]. Exponential Linear Unit (ELU) is used as the activation functions for operations instead of ReLu [5, 11]. This allows for a particular feature to be selected from a particular after convolution and activation operations are executed. Once completed, MaxPool is used for the reduction of the number of parameters and extract more features [21].

Temporal Feature Extractor The temporal feature extractor comprises of a 2 layer LSTM and an extended self-attention model. The LSTM network is used to learn features from EEG signals through the reliance of temporal data. The LSTM unit as shown in Fig 2.19 receives the output of the CNN for the current time Q_i, the previous output c_{i-1} feature and its hidden state h_{i-1}. This is then used alongside the activation functions of sigmoid and tanh to learn spatial and temporal features through the input and forget gate [7, 21]. This methodology is applied with the LSTM units in this model as they are corespondent to the number of EEG samples and the output generated from each time period is considered as an extracted temporal feature from each sample. Due to the existence of the second layer used to store all areas containing spatial and temporal features, the final output generated by the LSTM network are the hidden states of the second recurrent layer.

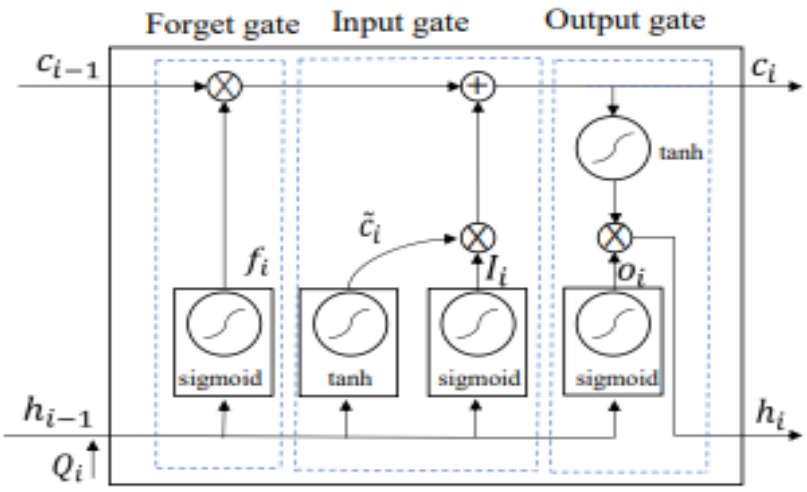

Fig. 4: LSTM unit architecture.

Figure 2.19: LSTM unit in the model

The extended self-attention mechanism is implemented with the aim to find more discriminative temporal information by finding the natural importance of each EEG sample being examined [6, 15]. This is done by understanding the similarity that every sample has from different areas using the findings from the hidden state h'_i. This can be seen in (49):

$$z'_i = f(h'_i, q_i) = W^T \sigma(W_1 h'_i + W_2 q_i + b_i) + b \qquad (49)$$

with z_i representing the intrinsic similarity for sample i with q_i being generated based on h'_i. Activation functions of σ and ELU are used with W and b used as weight and bias terms for the sigmoid function.

The probabilities of all EEG samples are then generated with each sample's probability being defined as :

$$p_i = \frac{exp(z_i'^T) \cdot h'_i}{\sum_{i=1}^{n} exp(z_i'^T \cdot h'_i)} \tag{50}$$

The final extended spatiotemporal attentive features for each sample is generated with the multiplication value of the probability of the sample and its respective hidden state. This can be seen from (51):

$$A_i = p_i \cdot h'_i \tag{51}$$

The classification method involved with the ACRNN by *Wei Tao et al* apply the spatiotemporal features [21] **A** with the softmax classifier to recognise the emotion of the participants as seen in (52):

$$P = softmax(WA + b) \tag{52}$$

where a probability matrix is generated that are in tune with each EEG sample, with W and b acting as weight parameters.

The accuracy of the model is determined using cross entropy loss across the labeled data through:

$$\nu = -\sum_{i=1}^{n} \hat{Y}_i log(P_i) \tag{53}$$

with $\hat{Y}_i$ representing the EEG signal at index i.

Chapter 3

Research Methodology

In this section, we describe the proposed preliminary pilot method that is used to test the capture and 4 class classification of EEG signals and its changes during the movement of the limbs of participants. Its implementation in the network architecture was motivated by the investigation of ensuring that the data collection procedure as well as the sections of preprocessing, feature extraction and classification were applied correctly and with efficiency. The generation of the confusion matrix is used as an indicator of ensuring the flow of data is continuous within the model and the connection, collection and processing EEG signals follows the intended network architecture that will be used for future works.

3.1 Acknowledgements

This study follows the guidelines of Macquarie University and is approved with the ethical approval number: 5201800483. The code for the GUI of this project is available here : https://github.com/Prithivi2001/EEG-Signal-Capture-Program-GUI

3.2 Data collection procedure

The two major components involved with ensuring an efficient process of data collection for this model involve the EEG cap which is used for collecting human emotion as well as the graphical user interface which acts as a direct stimuli for the user to watch and for the model to use to collect EEG signals that will be processed throughout the model. The participants who were chosen for this project comprised of the development team members of this project.

Samples were collected from each participant from watching 3 videos, each 25 seconds long that had the movement of one limb where users were asked to imagine them imitating whatever they saw in the Graphical User Interface(GUI).

3.2.1 EEG cap

The EEG cap used for this project was mainly made with a focus towards looking high frequency areas without requiring too many electrodes. Therefore, with the standard documentation of OpenBCI, a 3D printer was used to make the EEG cap used for experimentation in this project with 18 electrodes focusing on high frequency output areas from the human brain [3] as shown in Fig 3.1 and used with participants as seen in Fig 3.2

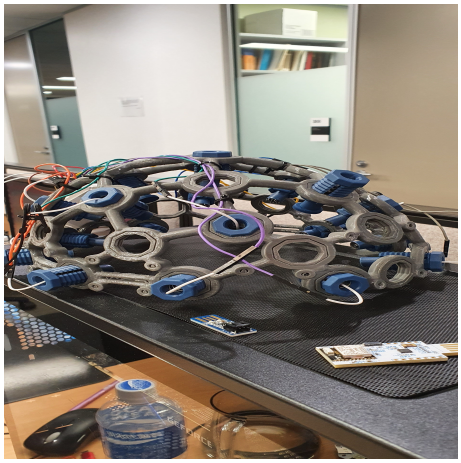

Figure 3.1: EEG Cap

Figure 3.2: Usage of EEG cap on participants

3.2.2 Graphical User Interface(GUI)

The GUI focuses on the display of videos for the 4 class movement classification and was developed using the TKinter library of Python. Earlier iterations of the GUI mainly focused on the display of images to generate a reaction from the users as seen in figures 3.3 and 3.4 . However, they were ineffective mainly due to the way they were presented, therefore resulting in the final iteration in Fig 3.5

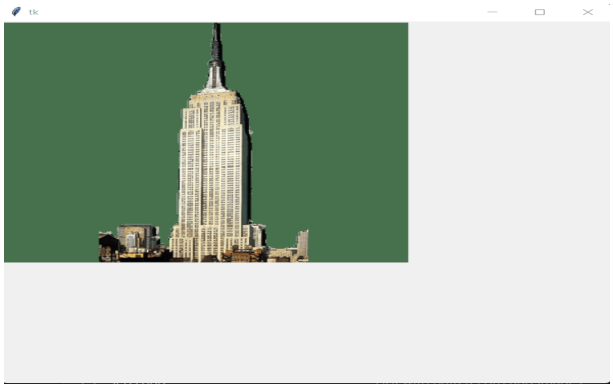

Figure 3.3: First iteration of the GUI

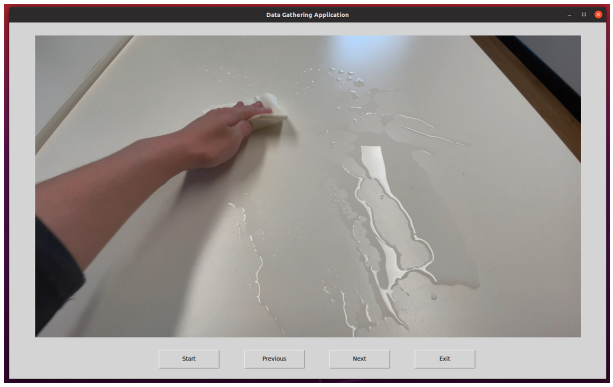

Figure 3.4: Second iteration of the GUI

Figure 3.5: Final iteration of the GUI

3.3 Network Architecture

3.3.1 Preprocessing

The preprocessing steps used in this model firstly divided the baseline signals every 25 seconds to separate the signals based on each video that the user has viewed and store that for feature extraction. Moreover, the signals generated every 5 seconds after every 25 seconds were removed as they were used as a buffer period to allow the user to go back to their original state in their brain [2]. This allowed for the concentration of users to improve and the classification of the participant to be calculated as well.

Filters like Band pass, Band stop, High pass and Low Pass were used as precautionary to remove unwanted noise and artifacts from the collected signals. SkLearn's normalisation that involved the scaling of input of vectors upon preprocessing was used to further refine the data before features were extracted. These filtering techniques were particularly used in order to be able to limit the signal to only focus on frequency areas which are investigated in this experiment via the EEG cap [3].

3.3.2 Feature Extraction and Classification

The feature extraction framework for this model involve the extraction of temporal features for the 3 time slots for each participant, each time slot representing one of the 3 videos that the user is watching during the experiment. As shown in Fig 3.6, the different time domain features that were extracted from the signals that are preprocessed include Mean, PTP, variance [6, 7, 18]. The feature are extracted and calculated from the the left and right side of the brain and concatenated upon calculation for use in classification using Random Forest classifier [6]. This allows for new understandings to be generated from the features that were generated from both sides of the brain.

The motivation behind the use of time domain features for the baseline model using a RF classification model was influenced from the data collection process and its application in the successful collection of features from areas of high frequency from the human brain from the onset of the planning process of this experiment [3]. Moreover, due to the framework set out for the experiment, the extraction of time domain features allow for a better understanding of the connections of different sides of the brain of the participant, how they interact with another and how influential those areas are in the movement of the participant's limbs during the time period of watching one video at a time, allowing for the classification process to be easier.

The classification process used with this model involved RF with the maximum level of depth being set to 5 and 10 samples of the featured data used for learning the data. This allowed for the network to better understand the features that have been extracted in order to make better classifications. Once the extracted features have been added to the classifier, the confusion matrix is generated for the testing data and the final classification values are generated.

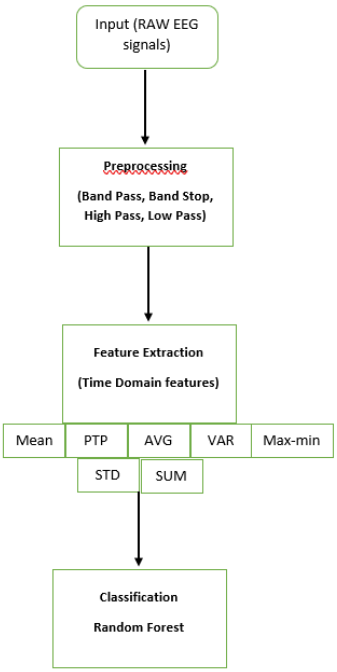

Figure 3.6: Pilot Method Research Methodology

3.3.3 Confusion Matrix Evaluation

The generated confusion matrix, as shown in Fig 3. below, highlight the accuracy of the model based on the input of RAW EEG signals of the participants of this project. As shown from the value of the true positive and true negative values, it is clear to see that the network architecture is fairly accurate for a pilot study when testing the capture and processing of EEG signals. The study can improve however, with a more improved model of an EEG cap that is capable of handling frequencies from other important areas and can be used as a starting point for introducing new models to improve the accuracy of the network.

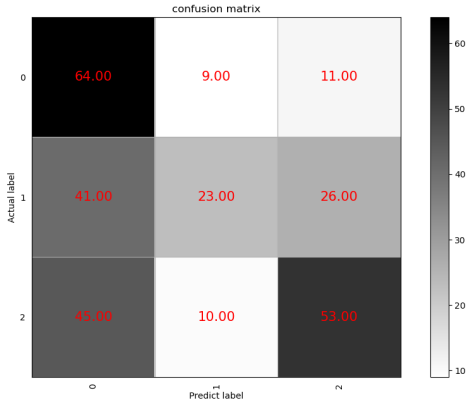

Figure 3.7: Confusion Matrix

3.4 Future Work

The future work for this project centre around the implementation of a convolutional self-attention framework for emotion classification. The main motivation behind the use of the components of convolutional networks is their ability to identify spatial and temporal relationships between captured signals from channels and being able to extract features much more efficiently in comparison to other models [6, 11]. In addition, the inclusion of the self-attention model can allow channels to be able to form relationships and identify areas which can lead to a better form of classification due to features between channels and frequency bands being identified more clearly [11, 16]. This model was also considered over other models as the use of other well-known models such as RNN has increased chances of vanishing gradient descent with long sequential data [7] and although it has been eliminated through the introduction of LSTM and GRU, RNN models still do not provide the same level of local relationships that self-attentions models do. The inclusion of CNN alone can also lead to problems of overfitting occurring with the data and exploding gradient when the model is being trained. The inclusion of self-attention along with the CNN allows for better learning of information between frequencies and electrode positions of frequency signals while also ensuring that the required features are extracted much more accurately as shown in the model [3, 11].

The next phase of this project will also look to be inclusive of 30-35 participants within the age groups of 17-23. This is mainly due to ease of access as this will allow the experimental team to be able to gather subjects much more quickly to further improve the model if needed. Another reasoning behind the demographic of the experiment being restricted to the age group highlighted earlier was because of the success of previous experiments that used EEG signals from this age group [13]. Moreover, being able to work with undergraduate students of this university can allow this experiment to also be able to highlight additional information that can give an indication to the emotional state of the participant [1, 17]. This in turn, can allow our model to be more effective in being able to not only detect emotions but other factors of mental health that the participant can be made aware of like depression and/or anxiety.

Additional changes will also be applied to the systems that are used to capture and classify EEG signals. The EEG cap that will be used to collect EEG signals in the future will follow the 10-20 standard and will allow for a stronger level of data collection when it will focus on the capture of signals that are mostly around high frequency regions of the human brain and will be applied to the conventional network architecture of the human brain. In addition, due to the type of model we are looking to implement, the use of different forms of features like time-frequency as well as more advanced classifiers like softmax can also allow for newer understandings and can come a long way in better improving the accuracy of our model [6, 16, 20].

Chapter 4

Conclusion

The understanding of emotional information is a focal point in allowing people to be informative on many different aspects. This report involved a deep level of study and analysis towards the different models and algorithms that are responsible for capturing and classifying EEG signals. This involved its examination in emotion classification as well as sleep stages [11].From understanding the benefits and drawbacks that are provided by each model, hand-made and state-of-the-art [6], we were able to highlight algorithms and models that are particularly accurate and useful in detecting and classifying human emotion which helped pave the way for deciding what type of model to use for the future, especially after applying its foundational methods successfully on another study.

Chapter 5

Abbreviations

EEG	Electroencephalogram
KNN	K Nearest Neighbor
SVM	Support Vector Machines
DBN	Deep Belief Network
CNN	Convolutional Neural Network(s)
RNN	Recurrent Neural Network(s)
LSTM	Long Short Term Memory
GRU	Gated Relational Unit
ReLU	Rectified Linear Unit
GUI	Graphical User Interface
DE	Differential Entropy
FFT	Fast Fourier Transform
ACRNN	Asymmetric Convolutional Recurrent Neural Network
RF	Random Forest
PCNN	Parallel Convolutional Neural Network
MASS	Montreal Archive of Sleep Studies

Bibliography

[1] A. Abdulrahman and M. Baykara, "A comprehensive review for emotion detection based on eeg signals: Challenges, applications, and open issues." *Traitement du Signal*, vol. 38, no. 4, 2021.

[2] F. Bahari and A. Janghorbani, "Eeg-based emotion recognition using recurrence plot analysis and k nearest neighbor classifier," in *2013 20th Iranian Conference on Biomedical Engineering (ICBME)*. IEEE, 2013, pp. 228–233.

[3] H. Becker, J. Fleureau, P. Guillotel, F. Wendling, I. Merlet, and L. Albera, "Emotion recognition based on high-resolution eeg recordings and reconstructed brain sources," *IEEE Transactions on Affective Computing*, vol. 11, no. 2, pp. 244–257, 2017.

[4] H. Chao and Y. Liu, "Emotion recognition from multi-channel eeg signals by exploiting the deep belief-conditional random field framework," *IEEE Access*, vol. 8, pp. 33 002–33 012, 2020.

[5] H. Cui, A. Liu, X. Zhang, X. Chen, J. Liu, and X. Chen, "Eeg-based subject-independent emotion recognition using gated recurrent unit and minimum class confusion," *IEEE Transactions on Affective Computing*, 2022.

[6] H. Cui, A. Liu, X. Zhang, X. Chen, K. Wang, and X. Chen, "Eeg-based emotion recognition using an end-to-end regional-asymmetric convolutional neural network," *Knowledge-Based Systems*, vol. 205, p. 106243, 2020.

[7] A. Garg, A. Kapoor, A. K. Bedi, and R. K. Sunkaria, "Merged lstm model for emotion classification using eeg signals," in *2019 International Conference on Data Science and Engineering (ICDSE)*. IEEE, 2019, pp. 139–143.

[8] F. P. George, I. M. Shaikat, P. S. Ferdawoos, M. Z. Parvez, and J. Uddin, "Recognition of emotional states using eeg signals based on time-frequency analysis and svm classifier." *International Journal of Electrical & Computer Engineering (2088-8708)*, vol. 9, no. 2, 2019.

[9] M. M. Hassan, M. G. R. Alam, M. Z. Uddin, S. Huda, A. Almogren, and G. Fortino, "Human emotion recognition using deep belief network architecture," *Information Fusion*, vol. 51, pp. 10–18, 2019.

[10] N. Jatupaiboon, S. Pan-ngum, and P. Israsena, "Emotion classification using minimal eeg channels and frequency bands," in *The 2013 10th international joint conference on Computer Science and Software Engineering (JCSSE)*. IEEE, 2013, pp. 21–24.

[11] Z. Jia, Y. Lin, J. Wang, R. Zhou, X. Ning, Y. He, and Y. Zhao, "Graphsleepnet: Adaptive spatial-temporal graph convolutional networks for sleep stage classification." in *IJCAI*, 2020, pp. 1324–1330.

[12] H. Jiang, R. Jiao, Z. Wang, T. Zhang, and L. Wu, "Construction and analysis of emotion computing model based on lstm," *Complexity*, vol. 2021, 2021.

[13] V. M. Joshi and R. B. Ghongade, "Idea: Intellect database for emotion analysis using eeg signal," *Journal of King Saud University-Computer and Information Sciences*, 2020.

[14] V. L. Kaundanya, A. Patil, and A. Panat, "Performance of k-nn classifier for emotion detection using eeg signals," in *2015 International Conference on Communications and Signal Processing (ICCSP)*. IEEE, 2015, pp. 1160–1164.

[15] D. Li, L. Xie, B. Chai, Z. Wang, and H. Yang, "Spatial-frequency convolutional self-attention network for eeg emotion recognition," *Applied Soft Computing*, vol. 122, p. 108740, 2022.

[16] G. Li, C. H. Lee, J. J. Jung, Y. C. Youn, and D. Camacho, "Deep learning for eeg data analytics: A survey," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 18, p. e5199, 2020.

[17] M. Li, H. Xu, X. Liu, and S. Lu, "Emotion recognition from multichannel eeg signals using k-nearest neighbor classification," *Technology and health care*, vol. 26, no. S1, pp. 509–519, 2018.

[18] T.-D.-T. Phan, S.-H. Kim, H.-J. Yang, and G.-S. Lee, "Eeg-based emotion recognition by convolutional neural network with multi-scale kernels," *Sensors*, vol. 21, no. 15, p. 5092, 2021.

[19] E. Salama, R. El-Khoribi, M. Shoman, and M. Wahby Shalaby, "A 3d-convolutional neural network framework with ensemble learning techniques for multi-modal emotion recognition. egypt. inform. j.(2020)."

[20] L. Shen, W. Zhao, Y. Shi, T. Qin, and B. Liu, "Parallel sequence-channel projection convolutional neural network for eeg-based emotion recognition," *IEEE Access*, vol. 8, pp. 222 966–222 976, 2020.

[21] W. Tao, C. Li, R. Song, J. Cheng, Y. Liu, F. Wan, and X. Chen, "Eeg-based emotion recognition via channel-wise attention and self attention," *IEEE Transactions on Affective Computing*, 2020.

[22] I. Wichakam and P. Vateekul, "An evaluation of feature extraction in eeg-based emotion prediction with support vector machines," in *2014 11th international joint conference on computer science and software engineering (JCSSE)*. IEEE, 2014, pp. 106–110.

[23] T. Zhang, W. Zheng, Z. Cui, Y. Zong, and Y. Li, "Spatial–temporal recurrent neural network for emotion recognition," *IEEE transactions on cybernetics*, vol. 49, no. 3, pp. 839–847, 2018.

[24] W.-L. Zheng and B.-L. Lu, "Investigating critical frequency bands and channels for eeg-based emotion recognition with deep neural networks," *IEEE Transactions on autonomous mental development*, vol. 7, no. 3, pp. 162–175, 2015.