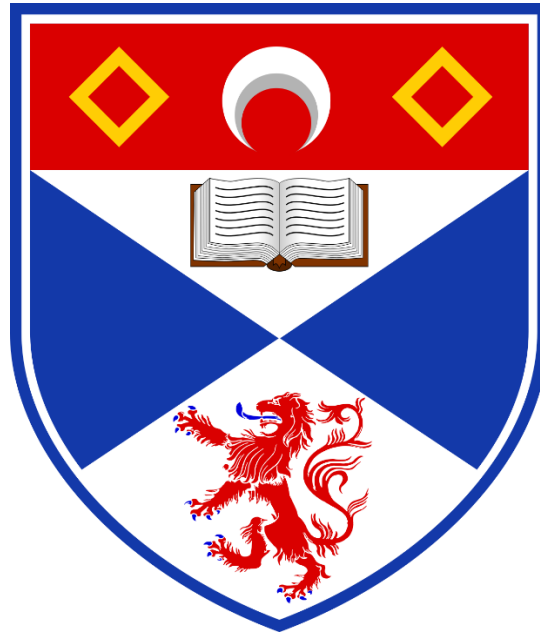


Deep learning model for EEG-based attention detection

Babs Khalidson



University of St Andrews

Supervisor: Dr Juan Ye

MSc Artificial Intelligence

August 14th, 2020

Abstract

Attention is at the core of neurological/cognitive functions, and deficits in attention have been linked to Alzheimer's disease (AD), Attention deficit hyperactivity disorder (ADHD), Traumatic Brain Injuries (TBI) and Posttraumatic Stress Disorder (PTSD). ADD can have severe consequences on a student's learning efficacy if it goes untreated. Traditionally, detecting inattentiveness is commonly done by observing an individual's expressions. However, this method is often inaccurate and increases the burden on teachers.

Interestingly, the detection of human attention levels can be automated with the use of deep learning (DL). Electroencephalography (EEG) signals provide a great source of information relating to human attention that can be analysed by deep learning algorithms. As a result, this study a novel deep learning architecture for EEG-based attention detection that builds upon the current state-of-the-art. The model predicted scores for attention, interest and effort on EEG data set of 18 users. Intra- and inter-subject classification results were evaluated using five-fold cross-validation. Results showed that the proposed model outperformed other deep learning and baseline models, where it was able to achieve an accuracy of 93% on a single user with binary classification

Declaration

“I declare that the material submitted for assessment is my own work except where credit is explicitly given to others by citation or acknowledgement. This work was performed during the current academic year except where otherwise stated. The main text of this project report is 14,442 words long, including project specification and plan. In submitting this project report to the University of St Andrews, I give permission for it to be made available for use in accordance with the regulations of the University Library. I also give permission for the report to be made available on the Web, for this work to be used in research within the University of St Andrews, and for any software to be released on an open-source basis. I retain the copyright in this work and ownership of any resulting intellectual property.”

Acknowledgements

I would like to pass my sincere gratitude to my supervisor Dr Juan Ye for her strategic support and guidance throughout the duration of the project.

I would also like to express my sincere gratitude to Mario Moreno Rocha for granting access to the EEG dataset and for his generous support for any data-related queries.

I would also like to thank Arkadiusz Kowalski for his support throughout the project. The discussions we had on how to tackle various areas of the project were crucial to its success.

Finally, I would like to thank my family for all the support they have provided me during my studies.

Table of Contents

Abstract.....	2
Declaration.....	3
Acknowledgements	4
Table of Contents	5
List Of Figures	8
List Of Tables	9
Chapter 1: Introduction.....	10
1.1 Motivation	10
1.2 Problem Definition	10
1.4 Objectives.....	11
1.3 Project Overview	11
Chapter 2: Context Survey.....	12
2.1 Introduction	12
2.2 Background	12
2.2.1 Attention.....	12
2.2.2 Electroencephalogram.....	13
2.2.3 Deep learning.....	13
2.3 EEG Analysis Methods.....	18
2.4.1 Overview	18
2.4.2 Data Preprocessing.....	18
2.4.3 Feature Extraction	19
2.4.4 Data Augmentation	20
2.4.5 Deep Learning Architectures	20
2.5 Summary	21
Chapter 3: Design.....	22
3.1 Introduction	22
3.2 EEG Dataset and Experimental Design	22
3.2.1 Overview	22
3.2.2 Experimental Design.....	23
3.2.3 Dataset.....	23
3.3 Data Pre-processing	24
3.3.1 Introduction	24
3.3.2 Window Sampling.....	24

3.3.2 Bandpass Filtering.....	25
3.3.3 Feature Scaling	25
3.3.4 Input Representation	25
3.4 Data Augmentation	25
3.5 Models.....	26
3.5.1 Baseline Models.....	26
3.5.2 Deep Learning Architectures	27
3.6 Version Control & Project management	30
Chapter 4: Implementation	31
4.1 Introduction	31
4.2 Dataset Creation	32
4.3 Model Implementation.....	32
4.3.1 Training	32
4.3.2 Hyperparameter Tuning	32
4.4 Predictions and generating results	32
Chapter 5: Evaluation	33
5.1 Introduction	33
5.2 Evaluation Design.....	33
5.2.1 5-fold Cross-Validation Within Users and Across Users	33
5.2.2 Evaluation Objectives	34
5.2.3. Evaluation Metrics	34
5.3 Results.....	35
5.3.1 Visualisation	35
5.3.2 Baseline Model Performance.....	39
5.3.3 The Effect of Data Pre-Processing.....	41
5.3.4 Data Augmentation	43
5.3.5 Deep learning Model Performance	45
5.3.6. Regularisation	48
5.3.7 Training Strategy	49
5.4 Discussion and Critical Appraisal.....	50
5.4.1 Visualisation	50
5.4.2 The Effect of Data Pre-Processing	50
5.4.3 Data Augmentation	50
5.4.4 Deep learning model performance	50
5.4.5 Regularisation and training strategy.....	51

Chapter 6: Conclusions.....	52
6.1 Introduction	52
6.2 Achievements	52
6.3 Future work	52
6.3.1 EEG as an image.....	52
6.3.2 GANs-based data augmentation.....	52
6.3.3 Cyclical learning rates.....	52
6.3.4 Class balancing strategies	53
6.3.5 Variant DL architectures	53
6.3.6 Variant bandpass filtering frequencies	53
6.4 Limitations	53
6.5 Concluding remarks	53
Chapter 7: Ethics	54
References.....	54
Appendices.....	60
A. User manual.....	60
B. Programming language and deep learning framework.....	61
B.1 Programming Language	61
A.2 Deep Learning Framework	61
C. Clean Data Set Information	63
D. Ethical Application Approval Letter.....	64
.....	64

List Of Figures

Figure 1: Visual representation of the attentional bottleneck of human vision. Image source [18].	12
Figure 2: Illustration of a feedforward neural network architecture.	14
Figure 3: A common CNN architecture in which convolutional layers, ReLUs, pooling layers and fully connected layers. Image source [9].	15
Figure 4: Diagram showing how RNNs previous outputs to be used as inputs while having hidden states. Image source [33].	16
Figure 5: EEG channel position of the 8 channel EEG helmets. Fp1= 1, Fp2 = 2, Fz = 3, Cz = 4, PO7 = 5, O1=6, O2=7, PO8=8, A1=Left ear, A2 = Right ear.	22
Figure 6: The Experiment layout	23
Figure 7: The general workflow of the proposed system	24
Figure 8: An example of EEG window sampling	24
Figure 9: Visual representation of the EEGNet model. Source [11].	28
Figure 10: Visual representation of the hybrid CNN-RNN model.	29
Figure 11: Image of contributions to master over time	30
Figure 12: Plot showing the Pomodoros completed during the project period	30
Figure 13: Flow diagram showing the data set creation process	31
Figure 14: Illustration of the process of 5-fold cross-validation.	33
Figure 15: Class frequency distribution plots of all the target labels across users in the multiclass dataset. A= Attention, B=Effort and C=Interest. Percentages in green shown the proportion of each class with respect to the whole dataset.	35
Figure 16: Correlation heatmap of all users with multi-class labels.	36
Figure 17: Correlation heatmap of User 1	37
Figure 18: EEG channel 1 plot for User 1's attention score over two samples (0.48 seconds). Coloured lines represent attention scores 1-5.	38
Figure 19: Order test: LGBM Multi-class classification performance across users and all labels.	42
Figure 20: Bar chart comparing Hybrid's accuracy with and without data augmentation.	44
Figure 21: Hybrid model vs baseline model (RF) classification performance on all labels.	46
Figure 22: Confusion matrices of all the DL models across users for the attention label. Each number represents a score for attention. Accuracies were evaluated on the multi-class dataset.	47
Figure 23: A) Bar graphs comparing Hybrid and EEGNet models's multi-class performance with and without early stopping after predicting within users on all labels. Without early stopping the models were left to train for 100 epochs B) Plot showing the training and validation loss curves with early stopping for EEGNet.	48
Figure 24: The effect of dropout on the Hybrid model. The performance was evaluated across users on the multi-class attention dataset.	49
Figure 25: Plot showing the effect of learning rate on the Hybrid model.	49
Figure 26: Graph Showing PyTorch's research adoption over time	62

List Of Tables

Table 1: Table showing the number of sessions per subject from the cleaned dataset.....	23
Table 2: EEG input representation for each model	25
Table 3: Classification accuracies of the baseline machine learning models after 5-fold cross-validation across all users. The accuracies highlighted in green are the maximum accuracies for each row.	39
Table 4: Binary and Multi-class classification performance of the baseline models within users for all labels (attention, interest, effort).	40
Table 5: Binary and Classification results across and within users for LightGBM and Random Forests..	41
Table 6: Multi-class classification performance of EEGNet across users and all on varying window sizes	41
Table 7: Multi-class classification accuracies of across users and within users (combined) for all labels on all DL models. Accuracies include with and without bandpass filtering.....	42
Table 8: Hybrid model accuracies from data augmentation test via multi-class classification on all labels across all users and within users	43
Table 9: Binary and multi-class classification performance of the DL models on all the labels. User 1 -18 show the performance within users while “across” highlights the performance across users	45
Table 10: DL models vs baseline model (RF) classification performance on all labels.....	46
Table 11: Comparison between PyTorch and TensorFlow	62
Table 12: Table showing the amount of EEG data per user	63

Chapter 1: Introduction

1.1 Motivation

According to recent studies, more students are going into higher education with learning disabilities. A common learning disability is Attention Deficit Disorder (ADD). ADD is usually characterised by a triad of symptoms such as hyperactivity, inattention and impulsivity. ADD can have severe consequences on a student's learning efficacy if it goes untreated. Traditionally, detecting inattentiveness is commonly done by observing an individual's expressions. However, this method is often inaccurate and increases the burden on teachers. Likewise, with the increased incidence of remote learning as a result of the 2020 COVID-19 outbreak, tools to detect the attention levels of students have been more needed. Failure to detect whether a student is inattentive or has ADD can limit the teacher's chances of providing additional educational support.

Interestingly, the detection of human attention levels can be automated with the use of deep learning (DL). Electroencephalography (EEG) signals provide a great source of information relating to human attention that can be analysed by deep learning algorithms. A study performed by Moreno Rocha *et al.* [1], recorded the EEG signals of 25 users while they read various types of academic papers either on paper or electronically. Attention, interest and effort scores were annotated to each paragraph after reading. In this dissertation, we analyse the EEG dataset generated by Moreno Rocha *et al.* and propose a novel deep learning approach for predicting a user's attention, interest and cognitive load while reading. The model serves as a proof-of-concept for a tool that could identify levels of attention while reading for ADD or inattentive diagnosis.

1.2 Problem Definition

Even though EEG has served as a valuable tool for analysing brain activity, it does come with limitations that hinder its effective analysis or processing. Most notably, EEG has a low signal-to-noise ratio (SNR) [2], [3] as the brain activity measured is often sequestered by other sources of environmental, physiological and activity-specific noise of similar or greater amplitude called artefacts [4]. Various filtering and noise reduction techniques such as bandpass filtering, have been used to minimise the effect of noise and to extract more nuanced brain activity from the recorded signals.

EEG is also a non-stationary signal [5], [6] that is, its statistics vary across time. As a result, a classifier trained on a limited amount of user data might generalise poorly to data recorded at another timeframe of the same individual. This is an essential problem for commercial EEG applications, which often need to work with a limited amount of data.

Moreover, the effectiveness of EEG applications can be limited by high inter-subject variability. This phenomenon arises due to the physiological differences between individuals, which vary in magnitude but can adversely affect the performance of models that are meant to generalise across subjects [7]. Since the ability to generalise between groups of individuals is important for many EEG applications, much research is being put into dealing with inter-subject variability.

To address some of the above-mentioned problems, this dissertation focuses on applying a new DL approach to simplify the processing pipelines significantly. DL allows automatic end-to-end learning of features and classification, while also reaching competitive performance on the task at hand. Indeed, in the last few years, DL architectures have been very successful in processing complex data such as images, text

and audio signals [8], leading to state-of-the-art performance on multiple public benchmarks—such as MNIST and Imagenet. [9], [10]. There are several ways DL can improve existing EEG processing methods. For instance, EEG feature extraction can be automated with the use of DL due to the hierarchical nature of DNNs. This finding has been proven by several studies [11]–[13] This also means features could potentially be learned on raw data, reducing the need for extensive pre-processing and feature extraction pipelines. Features learned through the DNN can also be more nuanced than manually crafted features. Moreover, DL facilitates the development of tasks that are seldomly performed on EEG data such as generative modelling [14] and domain adaptation [15]. Generative models can be used for data augmentation while domain adaptation allows the end-to-end learning of domain-invariant representations while preserving task-depended information. The focus of this study will be to use DL methods for feature extraction in order to speed up the processing pipeline.

1.4 Objectives

Below are the objectives that were outlined for the project:

Primary:

- Design a deep learning architecture for classifying EEG into scores of attention, cognitive load, and interest level
- Conduct a critical analysis of the classification results

Secondary:

- Create a variant deep learning design to help improve the classification results

1.3 Project Overview

Below are the chapters ahead and their brief overviews:

- **Context Survey:**
This chapter presents the background literature centred around attention, electroencephalogram and deep learning. This section also presents the literature review, where EEG-analysis methods where current EEG analysis methods are critically assessed.
- **Design:**
Here the experimental design and the design of the workflow are presented. Specifically, details on the data pre-processing, data augmentation, model design and version control are discussed.
- **Implementation:**
This chapter describes how the dataset was created, how the models were trained and how the results were generated.
- **Evaluation:**
This chapter presents the design of how the models were evaluated, the classification results and the discussion.
- **Conclusions:**
In this chapter, the achievements of the project are mentioned as well as future work and limitations of the project.

- **Ethics:**

This chapter presents the ethical considerations of the project.

Chapter 2: Context Survey

2.1 Introduction

This chapter focuses on providing the necessary context to the dissertation by providing an overview of the research topic and reviewing the current literature. We introduce the background and related works for EEG-based attention recognition.

Key terminologies relating to EEG and attention are discussed in the background. Related works in the field are critically accessed. Lastly, key methods in EEG analysis such as data processing, feature extraction and EEG classification are also discussed.

2.2 Background

2.2.1 Attention

Attention is the ability to focus on relevant stimuli while ignoring other stimuli in the peripheral environment [16]. Attention is a result of an attentional bottleneck, which is the amount of data the brain can process each second, for instance, human vision only accommodates 1% of visual input data [17] (**Figure 1**)

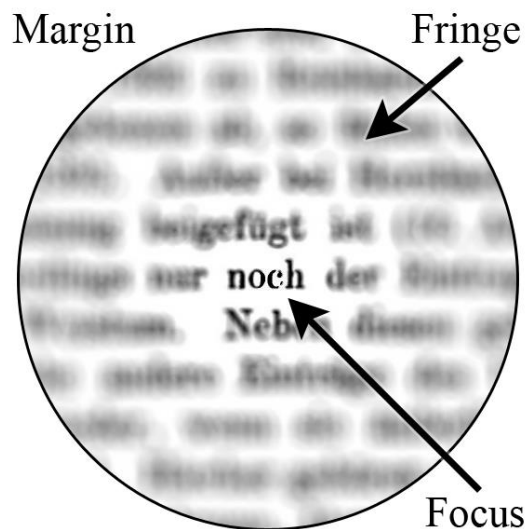


Figure 1: Visual representation of the attentional bottleneck of human vision. Image source [18].

Attention is at the core of neurological/cognitive functions, and deficits in attention have been linked to Alzheimer's disease (AD), Attention deficit hyperactivity disorder (ADHD), Traumatic Brain Injuries (TBI) and Posttraumatic Stress Disorder (PTSD) [19]. Moreover, an increasing number of students are entering post-secondary education with conditions like ADHD and tend to face the challenge of transitioning into a new environment [20]. The issue is more apparent in the case of distance learning, where the teacher is

unable to detect the attentional states of students remotely [21]. Therefore, building solutions that enable attention detection could be useful for ADHD diagnosis and remote measuring.

2.2.2 Electroencephalogram

Electroencephalogram (EEG) is a measurement of currents that flow during synaptic excitations of the dendrites of the pyramidal neurons in the cerebral cortex [22]. EEG signals are captured from certain locations over the scalp and normally measured in the time domain; however, they are subject to noise and artefact distortion, making the extraction of useful information a non-trivial task.

EEG signals can be divided by five frequency bands: delta (0.5–4 Hz), theta (4–7 Hz), alpha (8–13 Hz), beta (13–30 Hz) and gamma (> 30 Hz) [23]. Alpha waves are produced in the parietal and occipital regions of the brain when people are awake, quiet, calm, stable and focused. Beta waves occur when people are actively thinking and engaged in work. Beta waves occur in the frontal region of the brain and are apparent when a person is thinking, receiving sensory information, nervous or anxious. Theta waves primarily occur in the temporal and parietal regions of the brain and appear during drowsiness or deep physical relaxation. Delta waves are found during sleep and do not appear in normal adults who are awake. Some studies have found gamma waves to be related to selective attention, whilst others highlighted that gamma waves are associated with cognition and perceptual activity. [24]

Several types of biological signals, such as Electroencephalogram (EEG), electrooculogram (EOG), electrocardiogram (ECG), electromyogram (EMG), skin temperature variation and electrodermal activity, may be used to measure a human subject's attention level. Generally, electroencephalogram (EEG) is considered the most effective and objective indicator of attention level. EEG is also used because it is non-invasive, cost-effective and easy for recording [25].

2.2.3 Deep learning

2.2.3.1 Overview

DL, a subfield of machine learning is described as computational models that learn hierarchical representations of input data with multiple levels of abstraction [8]. Deep neural networks (DNN) or feedforward neural networks are stacked layers of artificial neural networks where each neuron applies a linear transformation to the input data they receive before feeding the result to a non-linear activation function. Importantly, the parameters of these transformations, such as the weights and biases, are learned by directly minimising the cost function. For instance, a classifier, $\mathbf{y} = \mathbf{f}^*(\mathbf{x})$ maps an input \mathbf{x} to label \mathbf{y} . A DNN defines the mapping $\mathbf{y} = \mathbf{f}(\mathbf{x}; \emptyset)$ and learns the values of the weights and parameters \emptyset that result in the closest function approximation [26].

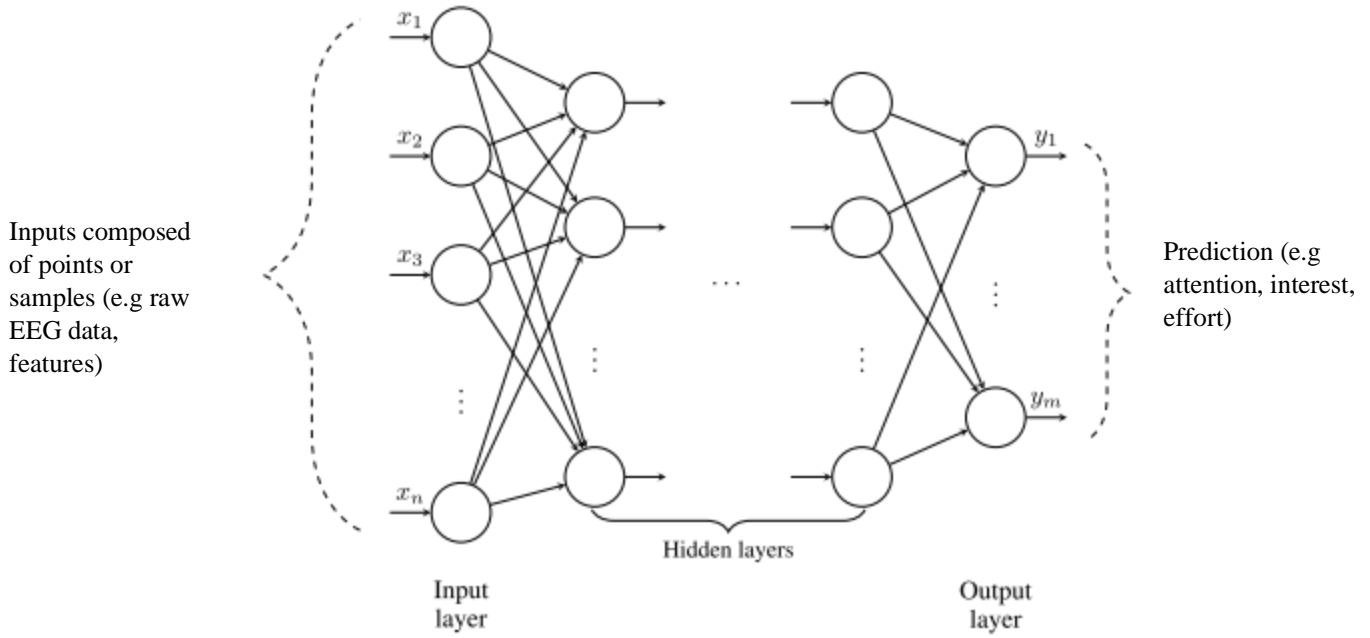


Figure 2: Illustration of a feedforward neural network architecture.

From **Figure 2**, we can see that the neural network is called **feedforward** because information is fed forward through the function being evaluated from \mathbf{x} , through the successive computations used to define \mathbf{f} , and finally to the output \mathbf{y} . There are no connections where the outputs of the model are feedback into itself. When a feedforward neural network includes feedback connections, they are called recurrent neural networks (RNNs), which are presented later in this **chapter**. Deep neural networks are called **networks** because they are represented by combining many different computations. For example, if we have three functions $\mathbf{f}^{(1)}$, $\mathbf{f}^{(2)}$ and $\mathbf{f}^{(3)}$ connected in a chain, all the functions are composed together to form $\mathbf{f}(\mathbf{x}) = \mathbf{f}^{(3)}(\mathbf{f}^{(2)}(\mathbf{f}^{(1)}(\mathbf{x})))$ [26]. In this example, $\mathbf{f}^{(1)}$ is called the **input layer** or the first layer of the network. $\mathbf{f}^{(2)}$ is called the second layer and $\mathbf{f}^{(3)}$ is the **output layer** or final layer. The term “deep” comes from the overall depth of the model, however, models can vary in depth depending on the data and task at hand.

During the training process, the model is shown the training data \mathbf{x} and produces an output in the form of a vector of scores that add up to 1. Each score represents a label, and we want the correct label to have the highest score. Each training example \mathbf{x} is associated with a label \mathbf{y} , and the output layer must produce a value close to \mathbf{y} . However, the behaviour of the other layers are not specified by the training data and are called **hidden layers** because the training data doesn’t show the desired output for these layers. The model computes a cost function that measures the error between the output scores and correct pattern of scores. The model then modifies the internal learnable parameters to reduce this error. The internal learnable parameters are usually the weights of the network and are real numbers that help determine the input-output mapping of the model.

To properly adjust the weights, the model implements an algorithm called backpropagation [27]. During backpropagation, information from the error flows backwards through the network, to compute the error gradient with respect to each weight. The model computes the gradients by applying the chain rule of derivatives and this gradient can be used to know how each weight can be tweaked to reduce the error. Next, the model implements gradient descent to adjust the weights in the opposite direction to the gradient

vector to minimise the error. The error averaged over the training samples can be seen as a hilly surface in high-dimensional space of weight values. The negative gradient vector indicates the direction of steepest descent on this surface, taking it closer to a global minimum, where the output error is low on average. This process is repeated continually until the network converges to an optimal solution.

Lastly, DNNs are called **neural** because they are loosely inspired by biological neurons. Biological neurons receive short electrical impulses called signals from other neurons via minuscule structures called synapses. When a neuron receives enough signals from other neurons, it fires its own signals. Individually, biological neurons behave in a simple way, but when organised in a vast network of billions of neurons, highly complex computations can be performed. Similarly, in an artificial neural network, each hidden layer of the network is vector-valued, and each element of the vector can be seen as acting analogously to a neuron. Each unit resembles a neuron by expressing a vector-to-scalar function that receives input from many other units and computes its own value after a certain threshold. The vector-scalar function is called an activation function and it is a non-linear function. The most popular activation function is the rectified linear unit (ReLU), which is expressed by: $f(\mathbf{z}) = \max(\mathbf{z}, 0)$. Other activation functions have been used in the past such as tanh or sigmoid activation functions, however, ReLU tends to learn faster in networks with many layers and reduces the incidence of the vanishing gradient problem [28].

Now that the concepts of DL and DNNs have been introduced, this dissertation will introduce the deep learning models used in this study; convolutional neural networks and recurrent neural networks.

2.2.3.1 Convolutional Neural Networks

Convolutional neural networks (CNNs) are a class of DNNs that were inspired by the animal visual cortex. They are used for analysing data that come in the form of multiple arrays or have a grid-like topology [26]. Grid-like data include time-series data which represent a 1D grid of samples at regular time intervals and image data, which considered as a 2D grid of pixels. The architecture of CNNs is comprised of an input layer, output layer and multiple hidden layers that are organised in a series of stages. The hidden layers are typically a series of convolutional layers that convolve – a specialised kind of linear operation. ReLU is commonly used as the activation layer and is followed by additional layers such as pooling layers, fully connected layers and normalisation layers.

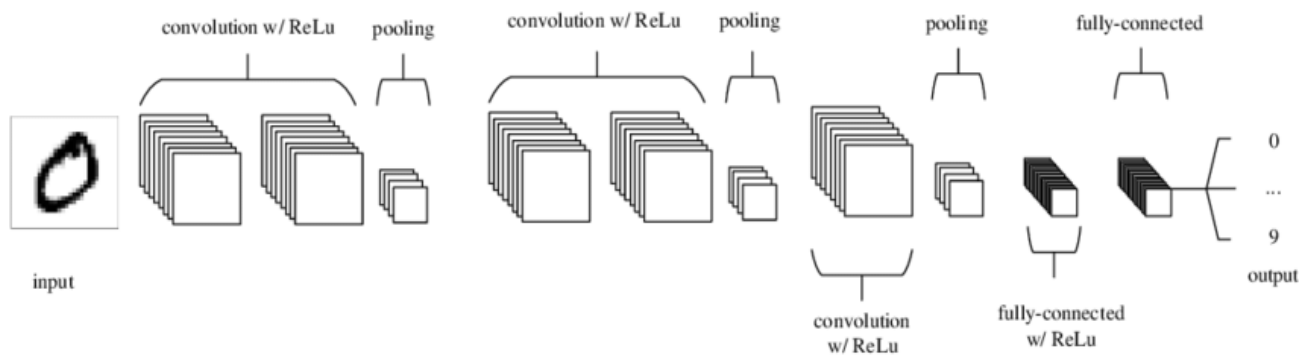


Figure 3: A common CNN architecture in which convolutional layers, ReLUs, pooling layers and fully connected layers. Image source [9].

The convolutional layers have neurons that are organised in feature maps, where each neuron is connected to local patches in the feature maps of the previous layer through a set of learnable parameters called filters or kernels. This hierarchical architecture allows the network to concentrate on small low-level features in the first hidden layer, then assemble them into larger higher-level features in the next hidden layer, and so on [29]. During the forward pass, each filter is convolved across the width and height of the input volume producing a 2D activation map of that filter through the ReLU layer.

Next, CNNs employ a pooling layer to merge semantically similar features into one [8]. Typically, pooling layers use max pooling, which computes the maximum value of local patch neurons in one feature map [30]. The pooling layer serves to reduce the dimensions of the representation progressively, to reduce the number of parameters, memory footprint and amount of computation in the network, which help control overfitting [31]. It is the norm to place a pooling layer after successive convolutional layers, with each followed by a ReLU activation layer. Finally, after several convolutional and max-pooling layers, the flattened matrix goes through a fully connected layer to classify the input. Backpropagation is also implemented as in a DNN, where all the weights in the filters are trained.

2.2.3.1 Recurrent Neural Networks

A recurrent neural network (RNN) looks very much like a feedforward neural network, except it also has connections pointing backwards [29]. This property makes RNNs recurrent in nature as it performs the same function for every input of data, while the output of the current input depends on the previous computation [32] (**Figure 4**). After producing the output, the output is copied and sent back into the recurrent network. For decision-making, it considers the current input and the output that it has learned from the previous input.

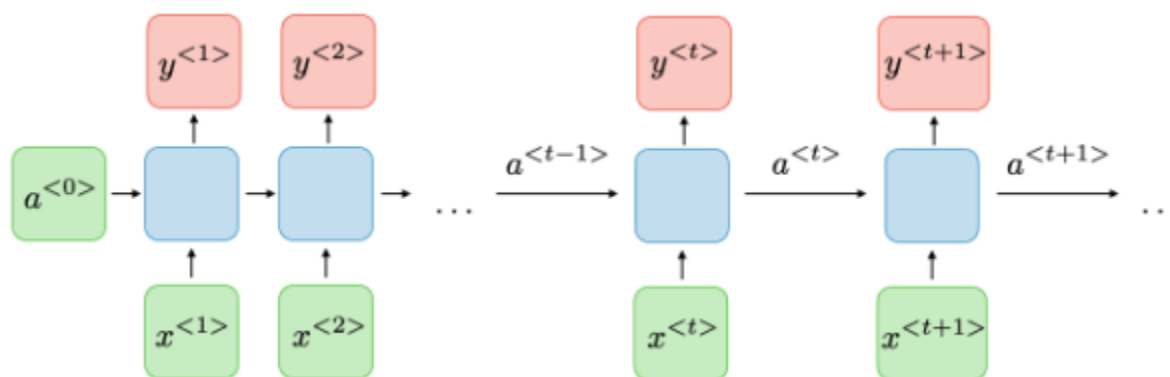


Figure 4: Diagram showing how RNNs previous outputs to be used as inputs while having hidden states. Image source [33].

Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. This makes them applicable to tasks such as time-series analysis, speech recognition, stock market prediction and connected handwriting recognition. This is due to RNNs having inputs that are related to each other while other neural networks have inputs that are independent of each other.

Despite RNNs ability to model sequential data, training an RNN can be difficult, and they are susceptible to the vanishing/exploding gradient problem [33]. Vanishing/exploding gradient phenomena occur because it is difficult to capture long term dependencies because of multiplicative gradients that are exponentially decreasing/increasing in relation to the number of layers [33].

To avoid the vanishing/exploding gradient problem, Hochreiter *et al.* [34] proposed a deep learning system called long short-term memory (LSTM). LSTM is usually augmented by recurrent gates called “forget gates”, which discover details that need to be discarded from the block. LSTM prevents error gradients produced during backpropagation from vanishing or exploding [35]. Instead, errors can flow backwards through unlimited numbers of virtual layers unfolded in space. That is, LSTM can learn tasks that require memories of events that happened a long time ago.

2.3 EEG Analysis Methods

This section reviews different methods used for EEG pre-processing, feature extraction and deep learning architecture design.

2.4.1 Overview

Over the years, previous research [36], [37] has shown that EEG signals contain important information about attention and cognitive load, suggesting the possibility of measuring an individual's attention level by observing these signals. As early as 1985, W.J Ray and H.W Cole [38] found a correlation between overall parietal alpha band and mental tasks in two experiments that were designed to examine the relationship between attentional effort and EEG during cognitive tasks of an emotional nature. In 1997, researchers analysed the power spectrum of EEG signals and found that it can reflect fluctuations in the level of alertness [39]. Yaomanee et al. [40] identified locations on the scalp that are suitable for detecting attention-related EEG signals. The results showed that α activity was slightly higher when the subjects were in a relaxed state, whereas β activity was greater when the subjects were attentive.

Several studies have analysed EEG signals to predict levels of attention using machine learning algorithms such as K -nearest neighbours (KNN), Support Vector Machines (SVM), Linear Discriminant (LD) classifiers and Bayesian classifiers. Li et al. [21] conducted EEG examinations using brain power-related tasks and asked the subjects to report their level of attentiveness. They employed KNN classification as the analysis method and designed a system for instantly measuring people's level of attentiveness. The classification accuracy of the system was 57.3%. The remaining artefacts mixed within the EEG signals may have contributed to the low accuracy. Liu *et al.* [41] investigated recognising the degree of human attention using EEG signals from mobile sensors and obtained a classification accuracy of 76.82% with an SVM classifier. Alirezai *et al.* [25] recorded EEG signals of 12 subjects for the binary classification of human attention using KNN, c-SVM, LD and Bayesian classifiers. Hamdadicharef *et al.* [42] conducted a study of recording EEG in various attention and non-attention states. Spectral spatial features from multichannel EEG were extracted and then classified using FLD to achieve an accuracy of 89.2%. Aliakbaryhosseinabadi *et al.* [43] classified EEG signals to identify variations in attention during motor task execution. They discovered that it is possible to explore a user's attention variation when performing motor tasks in a synchronous BCI system with temporal features. Beebe et al. [37] investigated whether sleep disorders during puberty cause inattention and consequently affect learning. The results indicated greater θ activity when the subjects were inattentive. Hu *et al.* [44] recognised attention by using correlation-based feature selection and KNN.

Few studies have deployed deep learning architectures for EEG-based attention detection. Interestingly Deckers *et al.* deployed two convolutional neural networks (CNN) to detect an attended speaker and the locus of auditory attention (left or right) in a multi-speaker environment [45]. Borhani *et al.* [46] conducted a study with 38 volunteers to decode EEG-based attentional state using CNNs.

2.4.2 Data Preprocessing

Rao *et al.* [47] deployed Independent Component analysis (ICA) and the Matching Pursuit (MP) algorithm for EEG processing. ICA is a mathematical tool for decomposing a mixed-signal into its statistically independent components. ICA can decompose multichannel EEG assuming that the measured signal is a linear mixture of several independent sources in the brain. In contrast, MP decomposes the EEG into several components by selecting from a dictionary of Gabor signals. The dictionary signals with higher correlation with the input EEG are chosen and subtracted from it. This process is repeated on the residues until the input EEG signal is satisfactorily represented by dictionary components.

For noise removal, several studies [21], [25], [48] removed frequency noise with the use of a bandpass FIR filter with cut off frequencies of 0.4HZ and 40HZ. Moreover, Hu et al. [44] used FastICA as an initial stage of de-noising as it has been shown to be effective in delineating overlapping frequency bands

For creating a standardised preprocessing pipeline, Bigdely-Shamlo et al [2], proposed PREP – a standardised processing procedure for large scale EEG analysis. The procedure is as follows: (i) Remove line-noise without committing to a filtering strategy. (ii) Robustly reference the signal relative to an estimate of the "true" average reference. (iii) Detect and interpolate bad channels relative to this reference. (iv) Retain sufficient information to allow users to re-reference using another method or to undo interpolation of a particular channel.

EEGlab a software for EEG analysis has plugins for data processing. Mognon *et al.* [49] used ADJUST, which uses ICA to remove features associated with various stereotypical artefacts, particularly eye blinks, eye movements, and discontinuities. Nolan et al. [50] used the FASTER pipeline, which uses a combination of statistical thresholding and ICA to identify and remove contaminated channels and bad epochs, particularly those associated with eye movements, muscle artefacts, trends, and white noise. Lastly, Lawhern *et al.* used DETECT, which uses [51] machine learning techniques based on auto-regressive features to identify artefacts of various types including eye movements, jaw clenching and other muscle artefacts.

2.4.3 Feature Extraction

EEG feature extraction is one of the most demanding steps of the traditional EEG processing pipeline [52], and the main goal of this dissertation is to get rid of this step by deploying DNNs for automatic feature learning. Indeed, Zhang *et al.* [13] deployed DNNs such as RNNs and CNNs for automatic temporal and spatial feature extraction of EEG data, respectively. Ren *et al.* [53] applied convolutional deep belief networks (DBNs) to the feature learning of EEG data and found that the learned features had better performance when compared to the current state-of-the-art feature extraction techniques. More recently, studies by Hartmann *et al.* [54] and Sturm *et al.* [55] were able to get promising results without manually extracting features.

Conversely, there are still many EEG studies that use manually crafted features. To name a few, Alirezaei *et al.* [25] hand-crafted 168 features based on four brain wave frequency bands: gamma, alpha, theta and beta. Some of the features were generated from calculating the entropy, maximum power, energy, mean and peak-to-peak value of each of the frequency bands. Another study generated EEG features based on the time-frequency domain by using short-time Fourier transform (STFT) for detecting binary user-preference (like versus dislike) [56]. Similarly, Ruffini *et al.* [57] generated spectrograms from EEG data from each channel by using Fourier analysis (FFT) after detrending blocks of 1 s with a Hann window¹ (FFT resolution is 2 Hz). Lastly, Acharya *et al.* [58] proposed a feature extraction technique which performs Wavelet Packet Decomposition (WPD) on the EEG segments to obtain the wavelet coefficients at different levels and uses Principal Component Analysis (PCA) to determine the eigenvalues of these coefficients.

An interesting question is whether hand-crafted features do outperform features learned by DNNs. While the answer to this depends on many factors such as domain application, there are several studies that have observed EEG as a raw input outperform hand-crafted input features. For instance, recently proposed DNNs for seizure classification using raw EEG data outperformed baseline models, such as SVMs, which were trained on frequency-domain features [59]. Additionally, frequency-domain features, are very similar to the temporal filters learned by a CNN [4]. Indeed, these features are often extracted using Fourier filters which

¹ The Hanning window is a taper formed by using a weighted cosine

apply a convolutive operation. Therefore, using raw EEG as input may become the best approach as the evidence to support its success continues to grow.

2.4.4 Data Augmentation

Data augmentation is a technique used for artificially generating new data examples from existing training data [60]. This can be considered as a form for regularisation, as adding more training examples allows the use of more complex models while reducing overfitting. Data augmentation has also proven effective in other fields such as computer vision, where images are rotated, flipped and cropped to generate more training examples [61]. When done correctly, data augmentation can help models generalise better by improving accuracy and stability [62].

Multiple EEG studies have found data augmentation beneficial. Wang *et al.* [63], added Gaussian noise to their training data to obtain new examples. They tested this approach on two different public datasets for emotion classification - MAHNOB-HCI [64] and SEED [65]. They found that augmenting the SEED dataset improved their accuracy of LeNet [66] from 49.6% to 74.3% and the accuracy of ResNet [67] from 34.2% to 75% when compared with not augmenting the data. Augmenting the MAHNOB-HCI dataset also proved beneficial as the accuracy of their ResNet increased by 5%. Their best accuracy was obtained from adding noise with a standard deviation of 0.2 and by multiplying the data by a factor of 30.

A few studies have investigated the use of overlapping windows as a form of data augmentation [59], [68]. Kwak *et al.* [69] tested different shift lengths between overlapping windows (10ms to 60ms out of a 2s window) and found that smaller shift lengths improved performance significantly. Furthermore, Schirrmester *et al.* [4], used overlapping windows to augment their data and significantly improved their training efficiency by implementing "cropping training", which involved computing redundant chunks of samples only once.

Lastly, the problem of class imbalance can often occur during EEG analysis, and researchers have used data augmentation to tackle this problem. Indeed, Schwabedal *et al.* [70] targeted the class imbalance problem of low-occurring sleep stages by generating Fourier transform (FT) surrogates of raw EEG data on the CAPSLPDB dataset. They managed to improve the accuracy of some classes by up to 24%. Likewise, Vilamala *et al.* [71] randomly balanced all the classes of transitional sleep stages while sampling for each training epoch. Similarly, Chambon *et al.* [72] used a balanced sampling strategy to improve overall balanced accuracy of an EEG-based temporal sleep stage classification task.

2.4.5 Deep Learning Architectures

The choice of neural network architecture plays a crucial role in DL-based EEG analysis. A review by Roy *et al.* [4] assessed 154 papers that used DL for EEG analysis and found that 40% of the papers used CNNs, while 13% of the studies used RNNs and autoencoders (AE) respectively. Interestingly, combinations of CNNs and RNNs were used in 7% of the studies. The rest of the studies used restricted Boltzmann machines (RBMs), fully connected (FC) neural networks, DBNs and generative adversarial networks (GANs). A recent study [73] not included in the Roy *et al.*'s review, proposed a capsule network (CapsNet) for learning various properties of EEG signals to achieve better and more robust performance than previous CNN methods. This suggests that capsule networks may start to become a popular approach for EEG analysis.

DNNs are composed of many layers, and the optimal number of layers needed for EEG analysis is still undecided. A study found that most DL-based EEG studies have at most ten layers. This is much smaller than the architecture depths most commonly used for computer vision like ResNet-18 (18 layers) and VGG-16 (16 layers). Schirrmester *et al.* [12] proposed 3 CNN models for EEG analysis based on the number of convolutional layers - a "shallow" CNN with two layers, a "deep" CNN with five layers and a residual

network with up to 31 layers. The study found that both the shallow and deep CNNs had comparable performance. However, the residual network performed much worse, suggesting that shallower models can achieve better performance in certain contexts.

Moreover, specific choices regarding the architecture design were made to enable to extract EEG features effectively. For instance, max pooling is used over mean pooling, as it is used to produce invariant feature maps to slight translations on the input. A few studies [69], [74], [75] used one-dimensional convolutions in the input layer for processing either temporal or spatial information independently at this point of the hierarchy. One study [13] used RNNs and CNNs for temporal and spatial feature learning, respectively, combined the features with using a stacked and then fed the output of the data transformation to an XGBoost classifier. Moreover, dropout and batch normalisation have also been used for regularising the DNN and stabilising the learning process [12]

2.5 Summary

There has been significant work put into understanding the relationship between attention and EEG signals over the years; however, there are still many opportunities for improvement. Firstly, the majority of the EEG studies that use deep learning have predominantly used CNNs or RNNs however; fewer studies have investigated the use of hybrid models. Secondly, most of the machine learning-based EEG studies undergo a time-consuming feature extraction process which usually requires human expertise and could be sped up with the use of deep learning. Thirdly, a person has varying levels of attention; however, most studies have only carried out binary classification, i.e. attentive and non-attentive state. Lastly, very few studies have worked with larger than > 20 subjects; thus, the current literature comprises of studies with limited data.

Chapter 3: Design

3.1 Introduction

This chapter discusses the design of the workflow that was implemented for EEG analysis. This chapter will include a description of the dataset and the overall experimental design. Details around particular design choices for data pre-processing and models are also discussed with their accompanying justifications.

3.2 EEG Dataset and Experimental Design

3.2.1 Overview

For EEG-based attention, interest and effort classification, this study used the Instrumented Digital and Paper Reading dataset [1]. The dataset's researchers gave 25 participants 16 readings with five paragraphs each and recorded their EEG signals while they were reading. The researchers used Neuroelectrics² Enobio 8, an eight-channel, wireless EEG helmet with a sampling frequency of 500hz (**Figure 5**). They created their helmet using the International 10-20 scalp electrode placement system. Throughout the experiment, they calibrated the EEG signals for three two-minute pauses.

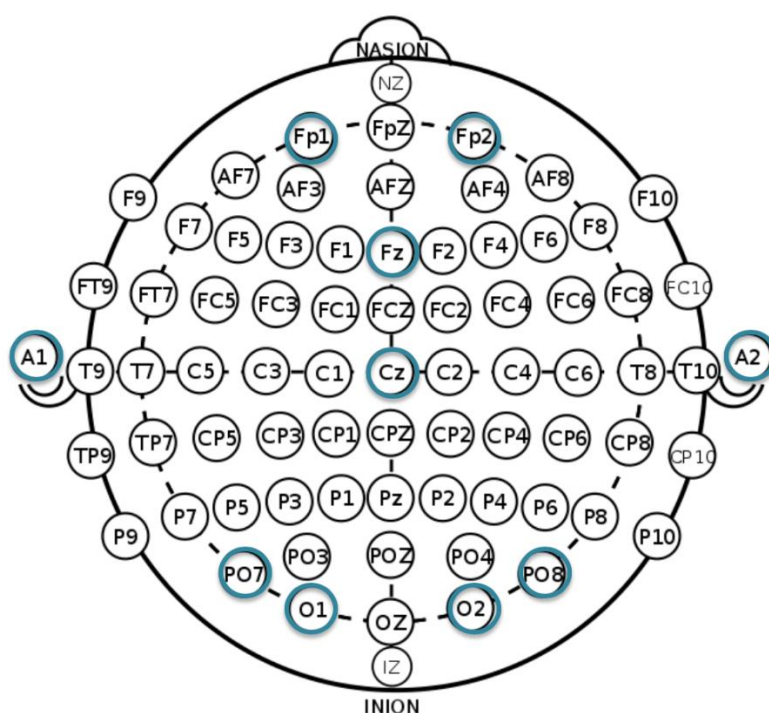


Figure 5: EEG channel position of the 8 channel EEG helmets. Fp1= 1, Fp2= 2, Fz = 3, Cz = 4, PO7 = 5, O1=6, O2=7, PO8=8, A1=Left ear, A2 = Right ear.

² Wireless EEG medical grade system for precise monitoring and mobile applications
<https://www.neuroelectrics.com/solutions/enobio/8/>

3.2.2 Experimental Design

The experiment took place in a room with two desks and two main stations- the experimenter³ and the participant stations⁴. (see **Figure 6**)

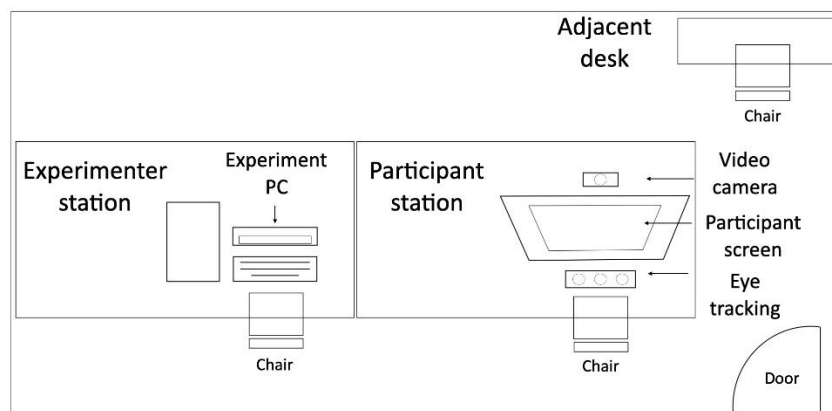


Figure 6: The Experiment layout

During the experiments, the experimenter asked participants to read a series of one-page documents written in English in a way consistent with how they would read texts for academic purposes; i.e., reading for comprehension. Participants had to signal to the experimenter when they had finished reading the document. At that point, they walked to a nearby table to finish a questionnaire, in which they scored each paragraph of the document they had just read in terms of how interesting they found the paragraph (Interest), how much attention they were paying to it (Attention), and how hard the paragraph was to read (Effort).

3.2.3 Dataset

The original dataset had 424 raw EEG data files that were generated from 25 participants. However, after cleaning the dataset, there were 18 users left with a total of 277 reading sessions (see **Table 1**).

Subject	1	2	3	4	5	6	7	8	9
# of Sessions	15	16	16	16	16	16	16	15	16

Subject	10	11	12	13	14	15	16	17	18
# of Sessions	9	15	16	16	16	16	16	16	15

Table 1: Table showing the number of sessions per subject from the cleaned dataset

The EEG CSV files had ten columns: columns 1 – 8 represented the EEG signals from the eight channels while columns 9 & 10 were the timestamps and the adjusted UNIX timestamp in seconds. Each session had an associated JSON file that contained the start and end timestamps for each of the five paragraphs, along with their scores for attention, effort and interest.

³ Experimenter’s station had a PC running Windows 10 and Dell 21” monitor running at 1680x1050

⁴ Participant station big screen with the Tobbi eye tracking, a video camera and Neurolectrics EEG helmet on the participant

3.3 Data Pre-processing

3.3.1 Introduction

The primary purpose of this study was to reduce the number of steps needed for EEG processing; hence the processing stage was kept to a minimum. However, based on the current literature [76], EEG processing steps such as bandpass filtering, window sampling and feature scaling were still necessary for effective DL-based analysis.

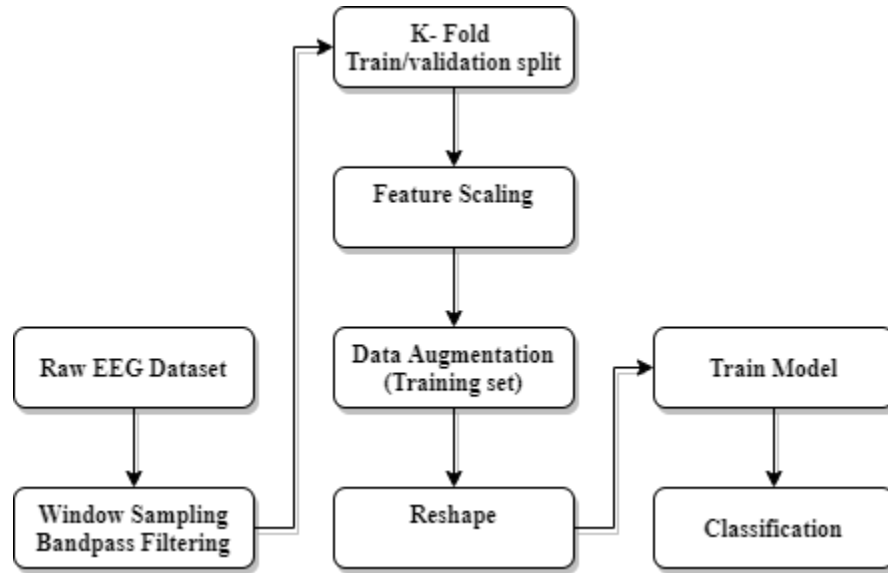


Figure 7: The general workflow of the proposed system

3.3.2 Window Sampling

The raw EEG dataset for each reading session had a shape of ($\# \text{timepoints}$, $\# \text{channels}$). To convert each dataset to a shape that was usable for the CNN, this study split the dataset into non-overlapping windows and converted the dataset into a 3D array of ($\# \text{samples}$, window size, $\# \text{channels}$). The window size was determined empirically after testing a range of values: [15, 30, 60, 120, 250]. This study found that a window size of 120 or 0.24s yielded the best performance and the results of the experiment are revealed in **section 5.3.3**

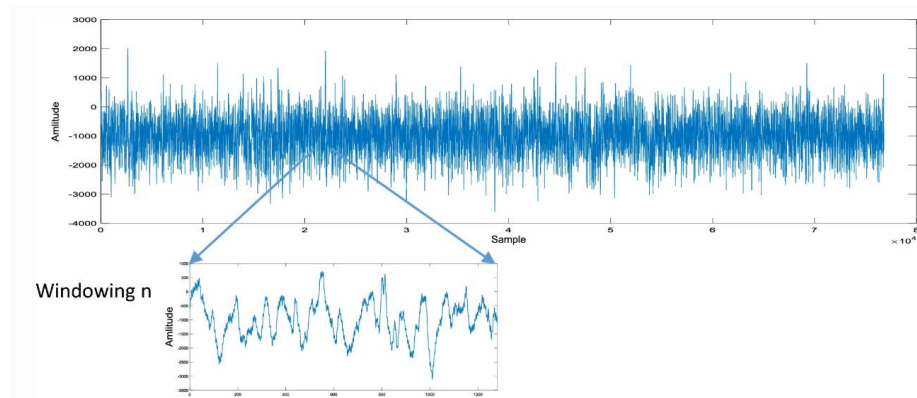


Figure 8: An example of EEG window sampling

3.3.2 Bandpass Filtering

Bandpass filtering is a standard signal processing method. The main function is to allow signals within a selected range of frequencies to be decoded while preventing signals at unwanted frequencies from getting through. Given EEG’s low SNR⁵, bandpass filters help improve the SNR by removing noise-related frequencies. Based on a study investigating EEG-based attention [41], this study applied low-pass and high-pass filters to limit the data between 0 – 40hz. Moreover, the order of the filter was set to 4. The stronger the order of the filter, the greater the attenuation and the higher the amplitude drop over the threshold frequency. This study determined the order value by testing different orders: [2, 3, 4,5,6] and picked the order with the highest accuracy after 5-fold cross-validation. (see **section 5.3.3**)

3.3.3 Feature Scaling

For scaling the data, this study computed the standard score for the training and validation sets.

The standard score of sample \mathbf{x} is calculated as:

$$\mathbf{z} = \frac{\mathbf{x} - \mathbf{u}}{\mathbf{s}}$$

where \mathbf{u} is the mean of the training samples, \mathbf{s} is the standard deviation of the training samples, and \mathbf{z} is the standard score.

3.3.4 Input Representation

Before the EEG data samples were fed into the models, they were reshaped based on the model architecture. For the baseline machine learning models, the raw EEG training samples were presented as 2D arrays, consisting of timepoints \times 8 channels. The Shallow and Deep CNNs were obtained from the Braindecode python library [12] and needed each batch to be shaped in 3D arrays of Batch Size \times Channels \times Timepoints. In contrast, the EEGNet CNN and CNN-RNN hybrid required a 4D input (see **Table 2**).

Model	Array Shape	Array Contents
Baseline ML Models	Timepoints \times 8	Timepoints \times Channels
Shallow and Deep CNNs	$32 \times 8 \times 120$	Batch Size \times Channels \times Timepoints
EEGNet and CNN-RNN	$50 \times 1 \times 120 \times 8$	Batch Size \times 1 \times Timepoints \times Channels

Table 2: EEG input representation for each model

3.4 Data Augmentation

This study considered using data augmentation to improve the accuracy and the robustness of classifiers. This study considered three different data augmentation techniques that have been used for image classification; geometric transformation, noise addition and GAN-based generation of samples. Geometric transformations such as rotation, reflection and shifting the data were not considered. The reason being was

⁵ Signal-to-Noise Ratio

because such transformations would destroy the temporal features of the EEG data [63]. GANs-based data augmentation was overlooked due to the size of the dataset as they would not have been enough samples per subject which would increase the risk of the model not converging. To avoid both problems, this study considered noise addition to augment the training samples. Noise can come in the form of Poisson, Salt, Gaussian and Pepper [77]. EEG signals are non-stationary and have a strong level of randomness, thus local noises like Poisson, Salt, and Pepper could alter the EEG features locally [63].

For these reasons, this study focused on adding Gaussian noise to each feature of the training samples. The probability density function P of a Gaussian random variable z is defined by [78]:

$$P_G(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}}$$

where z represents the grey level, μ is the mean value, and σ is the standard deviation. To ensure that there was no affect to the amplitude of the samples, Gaussian noise was generated with $\mu = 0$. To understand the effect of changing the standard deviation would have on the classifier, this study tested a range of values for σ between 0.001 and 0.8. Additionally, this study augmented the data by factors of 5, 20, 30 and 50.

3.5 Models

3.5.1 Baseline Models

Machine learning models commonly used for EEG analysis include random forests (RF), linear discriminant analysis (LDA) and LightGBM. This study trained these models on the raw EEG data to establish the baseline performance. In this section, this study introduces the main concept of each algorithm:

3.5.1.1 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) [79] is a common technique used for dimensionality reduction and classification. LDA provides class separability by drawing a decision region between the different classes. LDA tries to maximize the ratio of the between-class variance and the within-class variance. When the value of this ratio is at its maximum, then the samples within each group have the smallest possible scatter and the groups are separated from one another the most. [80]

3.5.1.2 Naïve Bayes

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable [81]. Bayes' theorem states the following relationship, given class variable y and dependent feature vector x_1 through x_n .

$$P(y | x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n | y)}{P(x_1, \dots, x_n)}$$

This study used Sklearn's [82] implementation of the Gaussian Naive Bayes algorithm for classification. The likelihood of the features is assumed to be Gaussian:

$$P(x_i | y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

The parameters σ_y and μ_y are estimated using maximum likelihood

3.5.1.3 LightGBM

Microsoft's LightGBM released in 2017, is a very popular powerful algorithm that is said to have a faster training speed than most algorithms, lower memory usage and better accuracy. It is a gradient boosting decision tree, unlike XGBoost, implements Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) to drastically reduce training times while still retaining a high accuracy [83]. GOSS helps to reduce the data size by removing significant proportions of data instances with small gradients while reduces the number of features by implementing a greedy algorithm [83].

3.5.1.4 Random Forests

Random Forests are an example of ensemble learning, where a group of Decision Trees are trained on a random subset of the training set [84]. The Random Forest algorithm introduces extra randomness when growing trees by searching for the best feature among a random subset of features as opposed to searching for the very best feature when splitting a node [85]. This results in a greater tree diversity, which trades a higher bias for a lower variance, generally yielding an overall better model.

3.5.2 Deep Learning Architectures

CNNs have had great success, automatically detecting and learning important features for image understanding and classification [86]. Inspired by this, recent studies [11], [12] have attempted to apply CNNs to EEG analysis and classification tasks. Moreover, "hybrid" CNN-RNN models have also been used for EEG analysis [13][87]. This section briefly describes the architectures of state-of-the-art CNNs for EEG analysis and a variant hybrid CNN-RNN model.

3.5.2.1 ShallowNet

ShallowNet [12] has a shallow network architecture with only two layers. The first two layers perform a temporal and spatial convolution. Temporal convolution is performed with 40 kernels whose dimension is 1×25 , and then spatial convolution is conducted with 40 kernels whose dimension is $E \times 1$, where E is the number of electrodes. The two convolutions are followed by a squaring nonlinearity, a mean pooling layer and a logarithmic activation function. Batch normalisation and dropout are applied to improve performance.

3.5.2.2 DeepNet

DeepNet, designed by the authors of ShallowNet, had four convolution-max pooling blocks. The first block was designed for EEG inputs while the next three blocks are standard convolution-max pooling blocks, and the final layer is a softmax classification layer. All layers except the dense layer use exponential linear units (ELUs) as activation functions [88].

3.5.2.1 EEGNet

This study developed a CNN architecture inspired by EEGNet [11]. EEGNet consisted of four building blocks (. The first block is learned with sixteen 2D convolutional kernels. In the second and third blocks, four 2D convolutional kernels with zero padding and 2D max-pooling are applied. The fourth layer is a classification layer with the Softmax function. Apart from the final dense layer, all the layers use the exponential linear unit (ELU) activation function. Batch normalization and dropout are applied to help improve classification results.

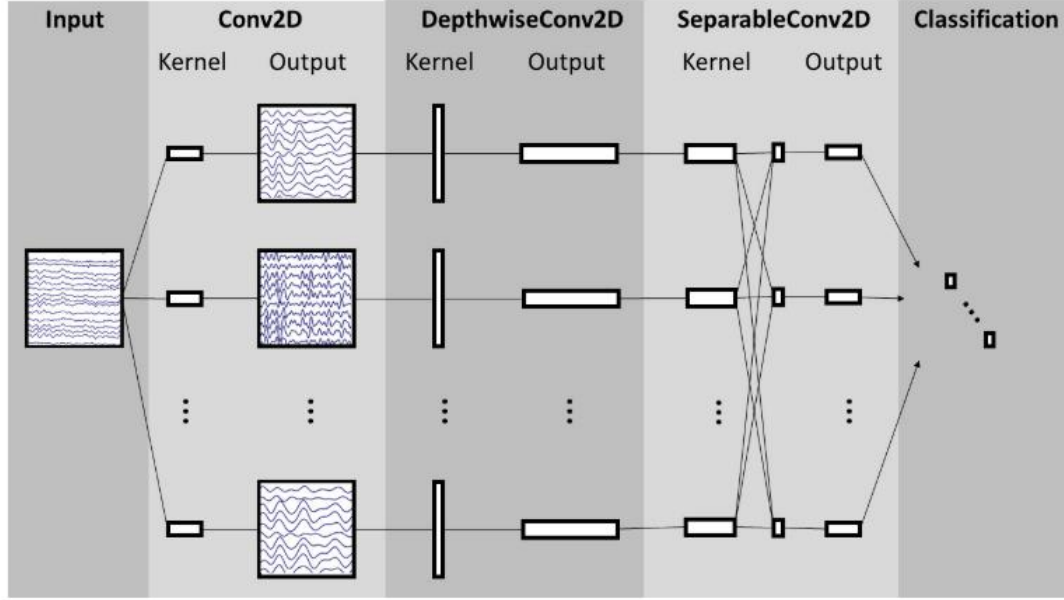


Figure 9: Visual representation of the EEGNet model. Source [11].

3.5.2.1 CNN-RNN (Hybrid model)

CNN-RNN (Hybrid model) was a composition of EEGNet and an LSTM RNN. The reasons why this study chose EEGNet were two-fold. Firstly, ShallowNet and DeepNet were both obtained from the Braindecode library, and they did not have direct compatibility with an RNN. Secondly, EEGNet outperformed ShallowNet and DeepNet in terms of classification results; therefore, it made sense to move forward with the top-performing CNN. Moreover, this project chose the LSTM RNN, mainly due to the temporal nature of EEG signals. Indeed, RNNs have the capacity for end-to-end temporal feature learning and classification; hence this study aimed to take advantage of this trait.

This study designed an RNN model with four layers: one input layer, one recurrent layer, an output layer and a softmax classification layer. The input size of the RNN was 56, and this was based on the input size of the final linear layer of the CNN. The hidden layer had sixteen hidden neurons and to prevent overfitting, this study determined the number of hidden neurons with this formula [89]:

$$N_h = \frac{N_s}{(\alpha * (N_i + N_o))}$$

N_i = number of input neurons.

N_o = number of output neurons.

N_s = number of samples in training data set.

α = an arbitrary scaling factor usually 2-10.

Based on the formula, the α can be used to decide the degree of generalisation. As mentioned in “*Neural Network design*” [90], limiting the number of free parameters in the model to a small portion of degrees

of freedom in the data can prevent overfitting. The degrees of freedom in the data is: $N_s * (N_i + N_0)$, thus changing α can prevent overfitting.

After designing the RNN, this study combined it with the EEGNet, by removing the fully connected layer of EEGNet, so that the learned features could be passed into the RNN. Once the features have been passed to the RNN, the RNN trains on the learned features before classifying using the softmax activation function.

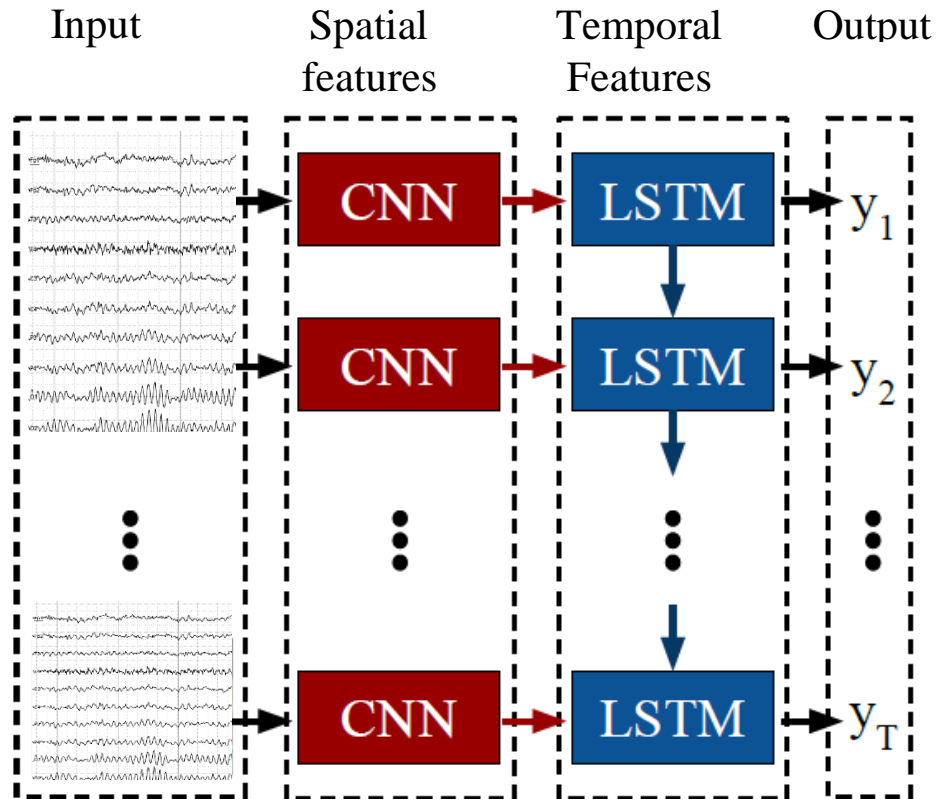


Figure 10: Visual representation of the hybrid CNN-RNN model

3.6 Version Control & Project management

Version control is a useful tool for keeping track of iterative code developments during a technical project. For keeping tracking of different versions of the project, this study used Github to push code frequently to an online repository (**Figure 11**). Mercurial was also used for keeping tracking of versions of the local repository.

For project management, this study used Microsoft Teams for team communication and meeting arrangements. A total of 10 meetings were conducted between Dr Juan Ye, Mario Moreno, Arkadiusz Kowalski and Babs Khalidson, to discuss the status of the project.

The Pomodoro Technique®⁶[91] was implemented daily to manage a consistent output of work.

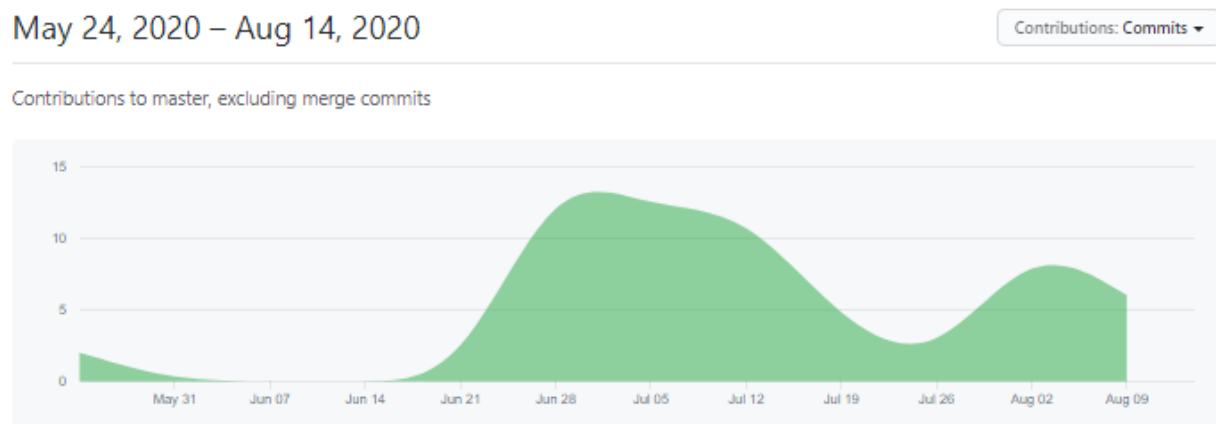


Figure 11: Image of contributions to master over time

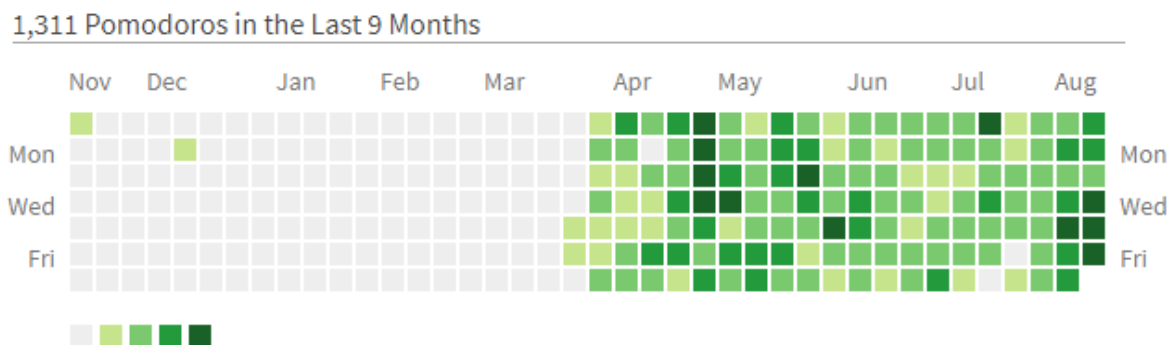


Figure 12: Plot showing the Pomodoros completed during the project period

⁶ The Pomodoro Technique is a time management method developed by Francesco Cirillo [91]. The technique uses a timer to break down work into intervals, traditionally 25 minutes in length, separated by short breaks.

Chapter 4: Implementation

4.1 Introduction

In this section, this study provides an overview of the technical implementation of the project. Here this study explains the dataset creation process, model implementation and how the results were generated.

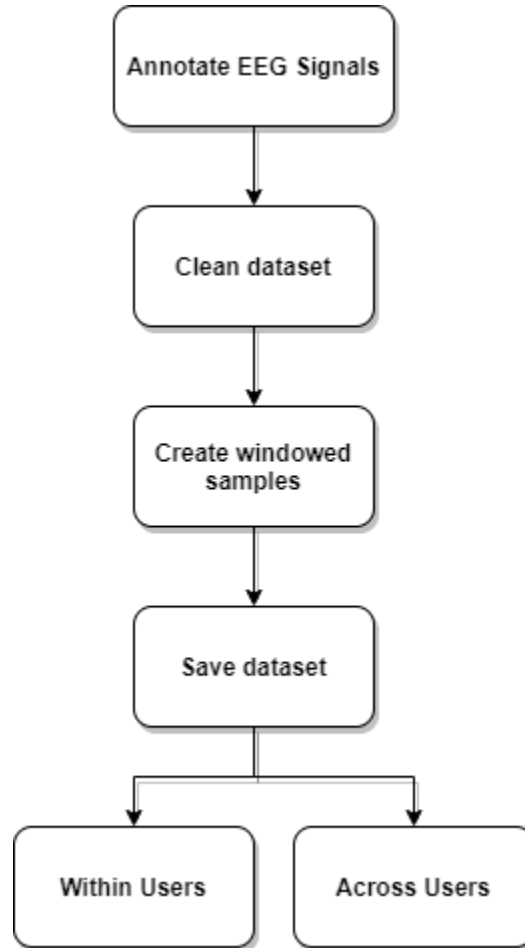


Figure 13: Flow diagram showing the data set creation process

4.2 Dataset Creation

To create the dataset (**Figure 13**), this study annotated each of the EEG CSV files using an associated JSON file that included the timestamps for different attention, interest and effort scores. Next, the dataset was cleaned by removing files with following criteria:

- Incorrect timestamp annotations
- JSON with no annotations
- Reading sessions with less than 5 paragraphs
- Duplicate timestamp annotations
- Users with a low amount of reading sessions after cleaning

Details on the exact amount of EEG data left per user can be found in **Appendix C**.

After cleaning the dataset, the data was then sampled with non-overlapping window sizes of 120 as described in **Section 3.3.2**. Finally, the dataset was saved as a dictionary either in the format for within user or across user analysis.

4.3 Model Implementation

This study used python and Pytorch for implementing the deep learning models. Justifications for their use can be found in **Appendix B**. ShallowNet, and DeepNet were obtained from the Braindecode library [92], while EEGNet and the Hybrid model were implemented in PyTorch. The ML models were built using sklearn.

4.3.1 Training

Training and testing were carried out for each model. During training, fivefold cross-validation (CV) was used, in which the EEG data obtained from the 18 users were analysed within users and across users. Details on how the fivefold cross-validation procedure was designed can be found in **Chapter 5.2.1**.

For the deep learning models, this study chose Adam [93] as the optimisation algorithm and cross-entropy as the loss function. A batch size of 50 was set for EEGNet and the Hybrid model while 32 was set for ShallowNet and DeepNet. Lastly, early stopping was implemented to reduce overfitting, where patience⁷ was set to 10 epochs.

4.3.2 Hyperparameter Tuning

This study ran a learning rate test on EEGNet and the Hybrid Model by increasing the learning rate exponentially between two boundaries and found that 0.001 was the optimal learning rate (**Figure 25**). Moreover, a range of dropout percentages was tested to observe the effect of dropout on the Hybrid model (**Figure 24**).

4.4 Predictions and generating results

This study exported the classification results in a CSV which recorded accuracy, precision, recall, f1 score, training time, window size and more useful metrics for analysis. Confusion matrices and loss plots were also generated after each run to serve as tools for debugging and analysis.

⁷ The number of epochs to wait before early stop if no progress on the validation set

Chapter 5: Evaluation

5.1 Introduction

This chapter explains the design of the evaluation process, along with the necessary justifications. Importantly, this chapter presents the results of the EEG visualisation, baseline model performance and deep learning architecture performance

5.2 Evaluation Design

5.2.1 5-fold Cross-Validation Within Users and Across Users

After cleaning the dataset, there were 18 users with each having around 15 reading sessions. This study assessed the performance of the models using 5-fold cross-validation. The reason why this study used this approach versus a simple train/test split was due to the high variance between the reading sessions across users and to have an approach that more rigorously assessed the ability of the models to generalise.

There were two different assessment types: within users and across users. For within users, this study evaluated the intra-subject performance of the models, i.e. ability to generalise per user based on each reading session. To do so, the 15 reading sessions of each user were randomly shuffled and then grouped into five folds. For example, $f1 = [3, 6, 10]$, $f2 = [8, 1, 5]$, $f3 = [0, 11, 7]$, $f4 = [4, 14, 9]$, $f5 = [2, 12, 13]$. Then each f was selected for validation and the rest for training. Thus, one iteration would look like, $f1$ for testing and $f2-f5$ for training. This was repeated iteratively through the five folds, and then the cross-validation performance was computed as the arithmetic mean over the five performance estimates from the validation sets.

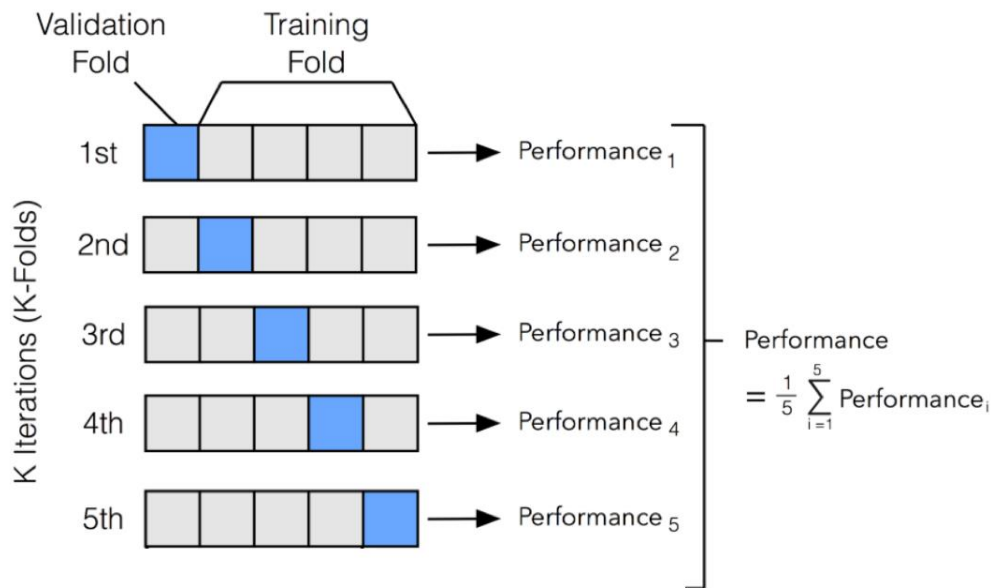


Figure 14: Illustration of the process of 5-fold cross-validation.

Similarly, 5-fold cross-validation was implemented for across users, where the inter-subject prediction performance was analysed. This time with 18 users, where each user had their reading sessions aggregated in one, and then the 18 users were split into five folds. i.e $f1 = [3, 6, 10]$, $f2 = [8, 1, 5]$, $f3 = [0, 11, 7, 4]$, $f4 = [14, 9, 2, 12]$, $f5 = [13, 16, 17, 15]$. Finally, the cross-validation performance was computed as described above.

5.2.2 Evaluation Objectives

This study had two evaluation objectives: multi-class and binary classification

- i) Multi-class classification: Each label (attention, interest, effort) had scores between 1-5, and the objective was to predict the current score based on the input.
- ii) Binary classification: This study converted scores of each label to either 0 or 1. Scores of 1 and 2=0 and scores 3,4,5=1. This was mainly done to see if there was any improvement in the classification accuracies by grouping the scores.

5.2.3. Evaluation Metrics

- Accuracy = $\frac{TP+TN}{TP+TN+FP+FN} \times 100$
- Precision = $\frac{TP}{TP+FP} \times 100$
- Recall/True-Positive Rate = $\frac{TP}{TP+FN} \times 100$
- Balanced Accuracy = average of recall obtained on each class.
- F1 Score = $2 \times \frac{Precision \times Recall}{Precision+recall} \times 100$

* For the multi-class datasets, the **macro** average was used for Precision, Recall and F1 score as it calculates the metrics of each label to find their unweighted mean

5.3 Results

5.3.1 Visualisation

Result 1: *The effort label had the highest degree of class imbalance across subjects*

This study visualised the class distribution of the target classes attention, effort and interest across the entire dataset. From the class frequency distribution plots below (**Figure 15**), we can see that there are significant cases of class imbalance for each of the labels. The effort label experienced the largest degree of class imbalance, where a score of 2 accounts for around 37% of the dataset, while scores of 5 only account for 7%. The interest dataset appears to be more balanced, where interest levels of [2,3,5] all having similar proportions at 20.91%, 21.43% and 20.95% respectively. While in the attention dataset, scores of 3 and 4 appear to dominate the dataset

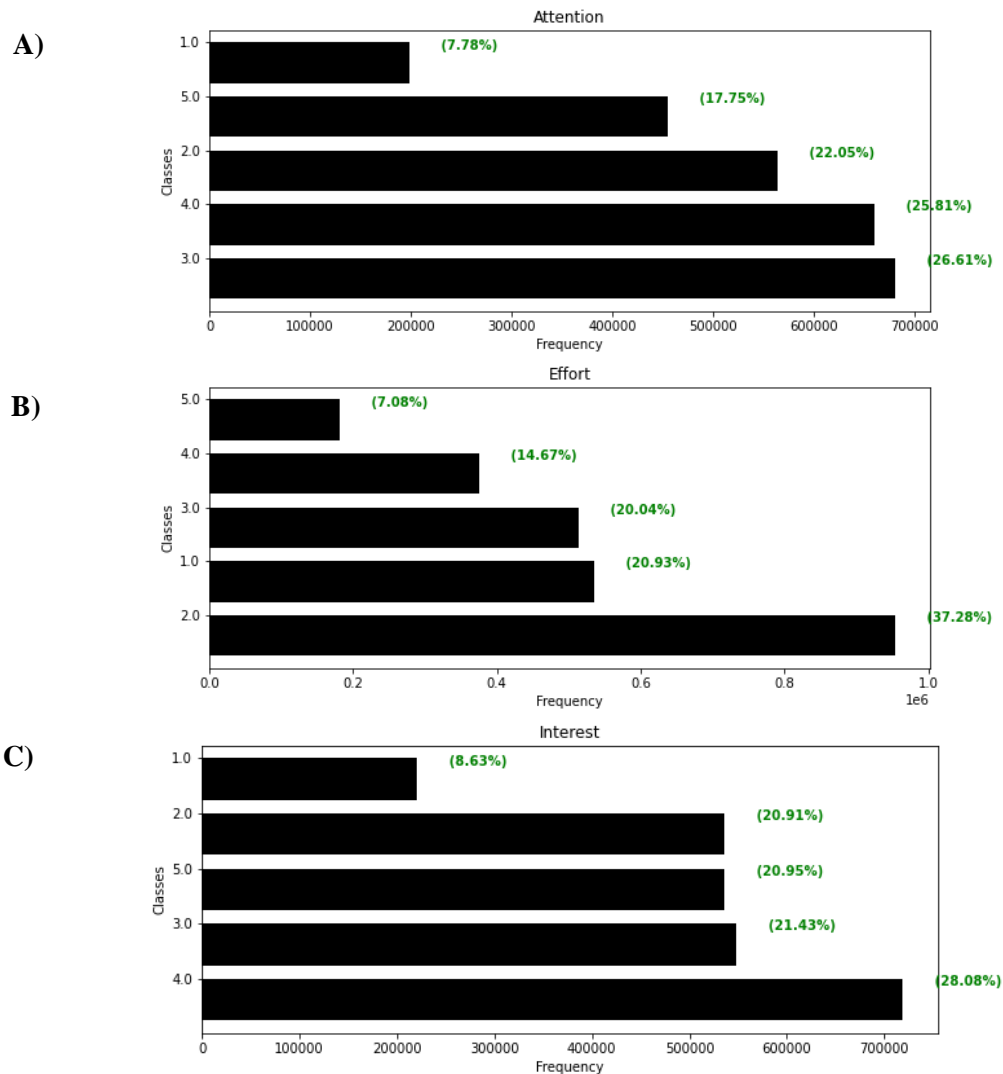


Figure 15: Class frequency distribution plots of all the target labels across users in the multiclass dataset. A= Attention, B=Effort and C=Interest. Percentages in green shown the proportion of each class with respect to the whole dataset.

Result 2: EEG data per user had a higher correlation with the target labels than the EEG data of all users

From the correlation heatmap of all users (**Figure 16**) we can see that the EEG channels have a negligible correlation with the target labels (interest, effort, attention), with values between -0.11 - 0.12. This confirms EEG's low signal-to-noise ratio and might indicate that signal preprocessing would be needed to help improve predictive performance. Interestingly the most correlation was seen between channels C7 vs C8 and C5 vs C6 with coefficients of 0.95 and 0.71 respectively. Attention and interest were shown to correlate the most with each other amongst the labels

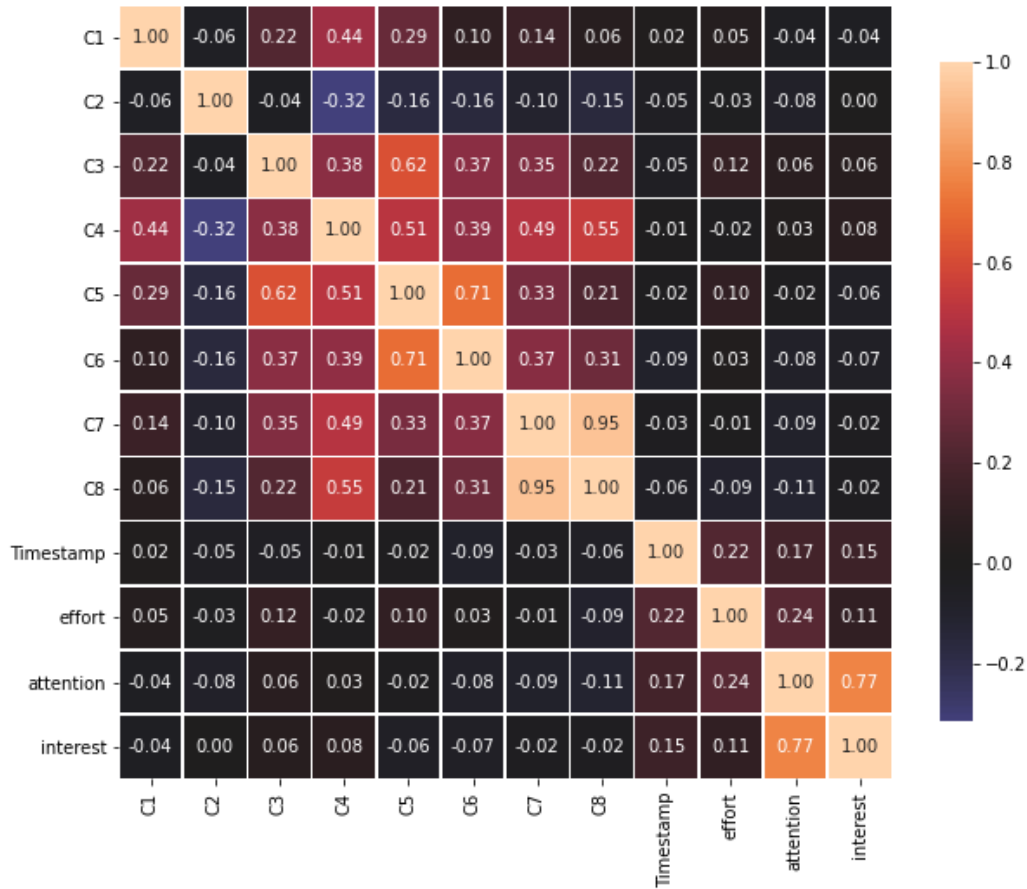


Figure 16: Correlation heatmap of all users with multi-class labels

On a per-user level (see **Figure 17**), the EEG channels of User 1 appeared to have higher positive correlations with the target labels than the aggregated EEG channels of all users, with values ranging from 0.07 to 0.33. Interestingly, the correlations between the channels were very high, with c1, c3-c8 all having correlation coefficients of either 0.99 or 1.

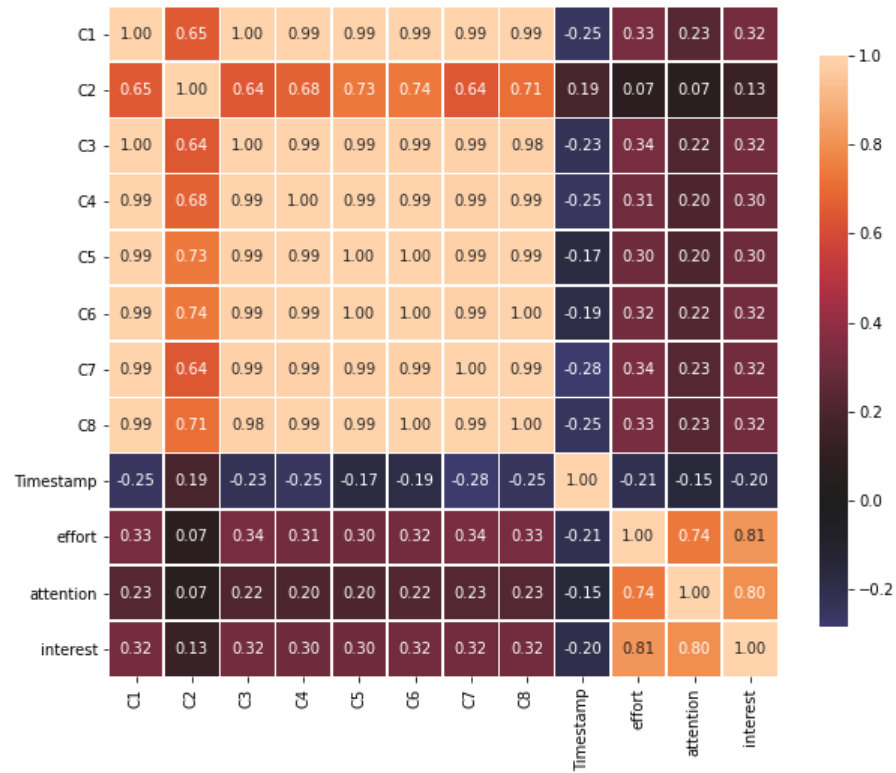


Figure 17: Correlation heatmap of User 1

Result 3: Amplitudes of attention scores 3,4,5 are similar while scores 1 and 2 are much lower

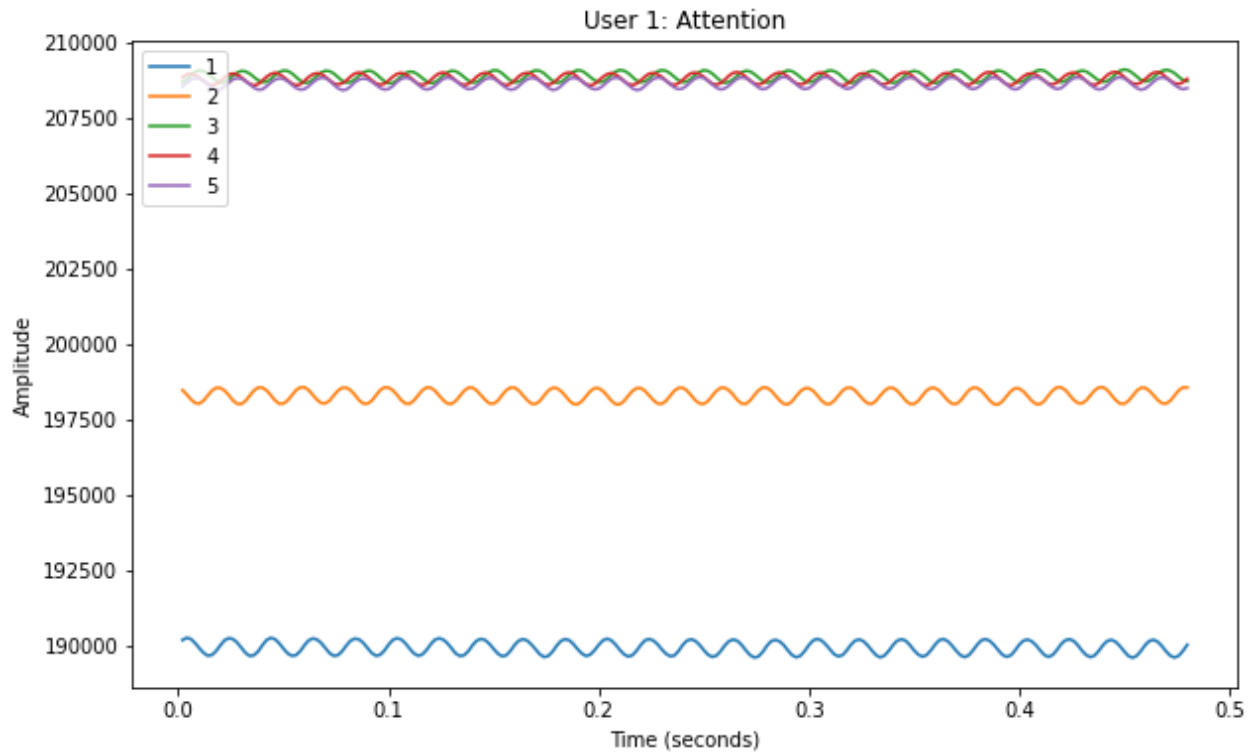


Figure 18: EEG channel 1 plot for User 1's attention score over two samples (0.48 seconds). Coloured lines represent attention scores 1-5

This study visualised the EEG signals of different scores of attention to have a better understanding of how the scores differed in amplitude. A subset of two samples (0.48 seconds) for each attention score (1-5) was drawn out of User 1's reading sessions. Channel 1 was used for visualisation because it had the highest correlation for attention in User 1 (see **Figure 18**). From the EEG channel plot, it is evident that the amplitudes for attention scores 3, 4, 5 are very similar with amplitudes of > 207500 . Attention scores of 1 and 2 are much lower but not as similar. Indeed there is a positive correlation between the amplitude of the EEG signals and the attention score, which confirms correlation coefficients seen in **Figure 17**. The similarity in 3, 4, 5 could affect the model's predictive abilities for multi-class classification. However, binary classification may be more promising than multi-class as the decision boundary appears to be clearer for differentiating scores [3,4,5] from [1,2].

5.3.2 Baseline Model Performance

Result 4: *LightGBM was the top-performing baseline model across users*

Label	LDA	LGBM	NB	RF
Binary	55.69%	57.77%	55.81%	58.79%
attention	55.62%	65.05%	48.10%	60.77%
effort	55.32%	52.84%	57.81%	50.18%
interest	56.15%	55.42%	61.54%	65.41%
Multi-class	24.61%	25.51%	21.30%	24.10%
attention	19.04%	19.69%	20.03%	21.53%
effort	32.44%	31.40%	24.95%	26.28%
interest	22.34%	25.44%	18.92%	24.49%
Average	40.15%	41.64%	38.56%	41.44%
Standard Deviation	17.03%	17.86%	18.90%	19.05%

Table 3: Classification accuracies of the baseline machine learning models after 5-fold cross-validation across all users. The accuracies highlighted in green are the maximum accuracies for each row.

This study ran all the baseline models with 5-fold cross-validation across all users and found that LightGBM was the top-performing model with an average accuracy of 41.64% (see **Table 3**), across all labels and classification types (binary and multi). LGBM's top performance appeared to be driven by its attention binary classification accuracy (65.05%) and having the highest multi-class classification on average (25.51%). Random forests came at a very close second, with an overall accuracy of 41.44%. This was driven by having the highest overall binary classification (58.79%) and having the highest accuracy for attention multi-class classification. LDA performed slightly worse with an overall accuracy of 40.15% but managed to have the highest accuracy for effort multi-class classification. Lastly, NB had the lowest accuracy out of all the models, with an accuracy of 38.56%. This was driven by the low accuracies seen on effort and interest multi-class classification and the low accuracy seen on attention binary classification.

Result 5: *Random forests were the top-performing baseline model within users*

Binary classification					Multi-class Classification				
User	LDA	LGBM	NB	RF	User	LDA	LGBM	NB	RF
1	0.51	0.66	0.49	0.66	1	0.21	0.2	0.27	0.18
2	0.81	0.82	0.84	0.83	2	0.29	0.22	0.31	0.25
3	0.7	0.71	0.78	0.69	3	0.26	0.27	0.28	0.28
4	0.53	0.56	0.46	0.55	4	0.27	0.31	0.31	0.36
5	0.55	0.65	0.6	0.61	5	0.31	0.4	0.42	0.39
6	0.47	0.52	0.58	0.67	6	0.21	0.37	0.27	0.34
7	0.73	0.74	0.64	0.73	7	0.24	0.28	0.25	0.33
8	0.42	0.47	0.48	0.53	8	0.24	0.18	0.19	0.18
9	0.4	0.51	0.45	0.52	9	0.23	0.25	0.2	0.18
10	0.63	0.41	0.37	0.34	10	0.46	0.27	0.28	0.22
11	0.86	0.88	0.83	0.91	11	0.48	0.44	0.46	0.52
12	0.59	0.57	0.61	0.63	12	0.28	0.31	0.25	0.28
13	0.56	0.66	0.65	0.69	13	0.31	0.47	0.36	0.55
14	0.55	0.56	0.66	0.61	14	0.22	0.27	0.25	0.27
15	0.74	0.85	0.83	0.86	15	0.32	0.37	0.32	0.41
16	0.62	0.58	0.51	0.75	16	0.21	0.23	0.22	0.25
17	0.68	0.67	0.54	0.65	17	0.2	0.3	0.27	0.31
18	0.52	0.56	0.53	0.61	18	0.31	0.32	0.26	0.38
Average	0.60	0.63	0.60	0.66	Average	0.28	0.30	0.29	0.32
StDev	0.13	0.13	0.14	0.13	StDev	0.08	0.08	0.07	0.11

Table 4: Binary and Multi-class classification performance of the baseline models within users for all labels (attention, interest, effort).

Within user, analysis showed that RF was the top-performing model on average for both binary (66%) and multi-class (32%) classification (see **Table 4**). RF's top performance overall was driven by having the highest accuracy for 9 out of 18 users on the binary classification and 10 out of 18 users on multi-class classification. RF's highest accuracy seen on the binary classification was 91% on user 11 while its highest accuracy on the multi-classification was seen on user 13 (55%). LGBM was the runner up in terms of performance with an accuracy of 63% on the binary classification and 30% on the multi-class classification. This was due to LGBM having the highest accuracy for 4 out of 10 users for both binary and multi-class classification. Interestingly, NB wasn't the worst performer this case, as it had an accuracy of 60% on the binary classification and 29% on the multi-class classification. NB outperformed LDA overall due to having the highest accuracies on the multi-class classification for 4 out of 18 users, while LDA only had the highest accuracy for 2 users.

Result 6: *Random forests outperformed LightGBM overall*

Despite LGBM having a higher performance across users, RF had a higher accuracy of 48%, after combining the binary, multi-class, within user and across user classification results (see **Table 5**).

Evaluation Type	LGBM	RF
Across Users	0.42	0.41
attention	0.42	0.41
effort	0.42	0.38
interest	0.40	0.45
Within Users	0.47	0.49
attention	0.48	0.5
effort	0.43	0.43
interest	0.50	0.52
Average	0.445	0.45

Table 5: Binary and Classification results across and within users for LightGBM and Random Forests

As a result, RF was chosen as the baseline model to compare against the top DL model

5.3.3 The Effect of Data Pre-Processing

Result 7: *A window size of 120 yielded the best performance*

Before training all the deep learning models, this study ran a test across users to determine the best window size. This study evaluated the multi-class classification performance of EEGNet across users on window sizes of 15,30,60 and 120. A window size of 120 proved to have the best performance overall with an accuracy of 24%. Interestingly, a window size of 15 had the joint highest balanced accuracy and recall; however, its training time was almost five times more than a window size of 120. As a result, a window size of 120 was chosen due to having the highest accuracy and the lowest training time.

Window Size	Run time (s)	accuracy	balanced accuracy	precision	recall	f1 score
120	25	24%	19%	17%	19%	14%
60	36	19%	17%	16%	17%	13%
30	104	21%	19%	14%	19%	13%
15	119	22%	19%	15%	19%	13%

Table 6: Multi-class classification performance of EEGNet across users and all on varying window sizes

Result 8: *Changing the order of the bandpass filters had a slight impact on performance*

Before applying the bandpass filters on all the DL models, I used one of the baseline models, LGBM to determine, which order value to use for the filter by evaluating its multi-class classification performance across users. Usually, the stronger the order of the filter, the greater the attenuation and the higher the amplitude drop over the threshold frequency. From the test, an order value of 4 appeared to have the highest accuracy (**Figure 19**); hence this was the value set for the bandpass filter.

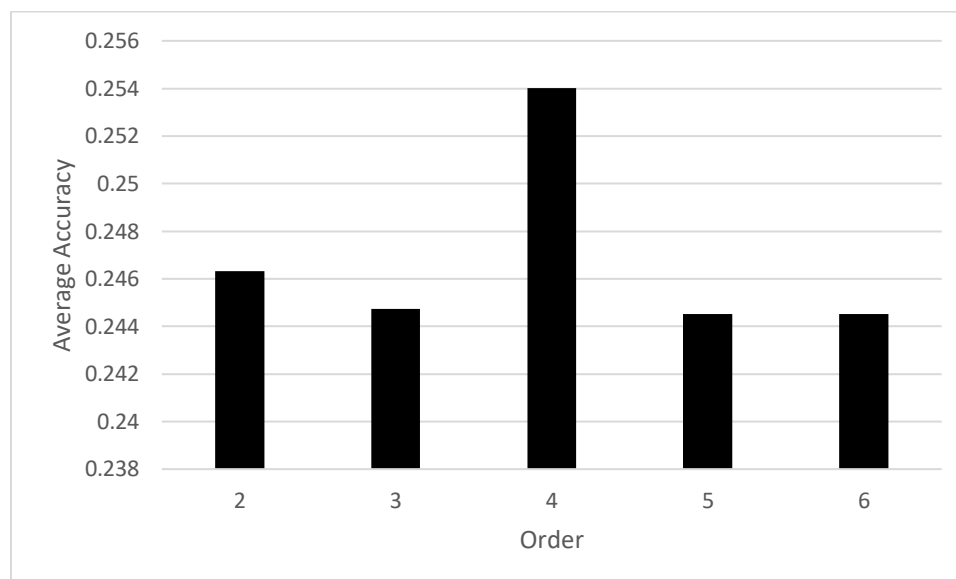


Figure 19: Order test: LGBM Multi-class classification performance across users and all labels

Result 9: *Bandpass filtering decreased performance for EEGNet and Hybrid*

After determining the order value for the bandpass filter, this study ran a multi-class classification test across users and within users to observe the effect of applying the bandpass filter on the DL models. The bandpass filter (0-40Hz) improved the performance of DeepNet and ShallowNet but decreased the performance of Hybrid and EEGNet. DeepNet saw the highest increase in accuracy as accuracy went up from 28.07% to 30.25%, while EEGNet saw the largest decrease.

Bandpass	DeepNet	EEGNet	Hybrid	ShallowNet
Without	27.86%	35.15%	37.50%	28.05%
With	30.04%	33.77%	37.43%	30.48%
Difference	2.18%	-1.38%	-0.08%	2.43%

Table 7: Multi-class classification accuracies of across users and within users (combined) for all labels on all DL models. Accuracies include with and without bandpass filtering.

5.3.4 Data Augmentation

Result 10: *Augmenting the data had different effects within users and across users*

This study augmented the data to help reduce overfitting and improve the performance of the hybrid model. Multi-class classification was used to evaluate the effect of data augmentation. From **Table 8**, it is evident that augmenting the data had different effects within users and across users. For within users, augmenting the data 20 times with $\sigma = 0.01$ yielded the best performance. While across users, augmenting the data only by five times with a high standard deviation of 0.5 led to the highest accuracy

Within Users				
Augmented Multiple	Standard deviation of Gaussian noise (σ)			
	0.001	0.01	0.1	0.5
5	0.386	0.392	0.387	0.375
20	0.398	0.399	0.394	0.395
30	0.381	0.384	0.389	0.390

Across Users				
Augmented Multiple	Standard deviation of Gaussian noise (σ)			
	0.001	0.01	0.1	0.5
5	0.270	0.280	0.280	0.285
20	0.268	0.264	0.262	0.247
30	0.243	0.263	0.254	0.258

Table 8: Hybrid model accuracies from data augmentation test via multi-class classification on all labels across all users and within users

Result 11: *Data augmentation improved accuracy within users but decreased accuracy across users*

After picking the best combinations of augmented multiple and standard deviation, this study compared the accuracy of the Hybrid model with and without data augmentation. Interestingly data augmentation only improved the accuracy of the model within users but didn't improve the accuracy across users (**Figure 20**).

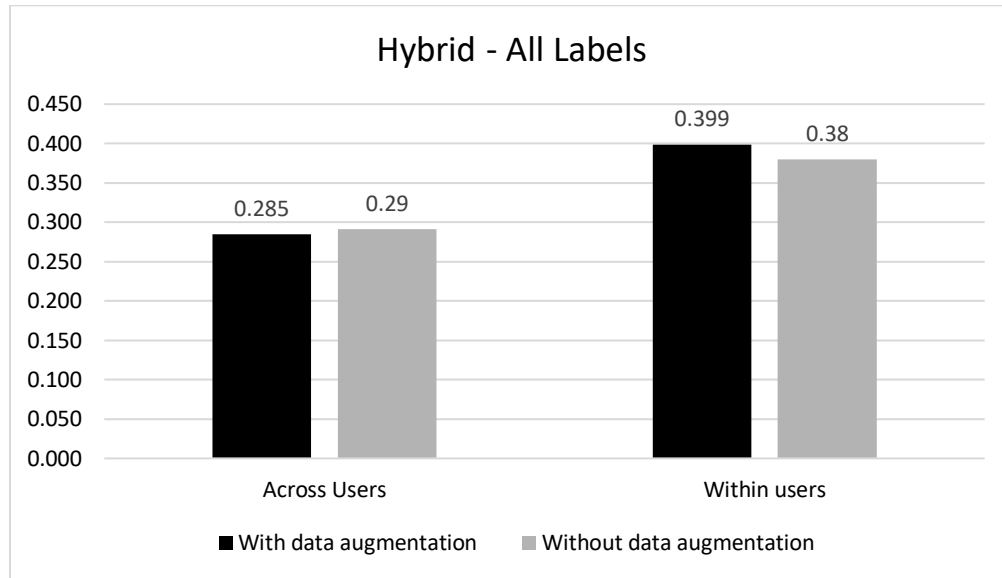


Figure 20: Bar chart comparing Hybrid's accuracy with and without data augmentation.

5.5.5 Deep learning Model Performance

Result 12: *The Hybrid model was the top-performing architecture*

Binary Classification					Multi-class Classification				
User	DeepNet	EEGNet	Hybrid	ShallowNet	User	DeepNet	EEGNet	Hybrid	ShallowNet
1	0.63	0.74	0.82	0.6	1	0.29	0.34	0.33	0.28
2	0.83	0.84	0.82	0.8	2	0.22	0.28	0.3	0.26
3	0.75	0.78	0.79	0.77	3	0.21	0.28	0.25	0.26
4	0.59	0.61	0.68	0.5	4	0.34	0.4	0.44	0.3
5	0.7	0.66	0.66	0.63	5	0.36	0.49	0.5	0.46
6	0.52	0.68	0.72	0.61	6	0.21	0.34	0.33	0.23
7	0.67	0.76	0.77	0.71	7	0.28	0.31	0.31	0.24
8	0.38	0.51	0.53	0.44	8	0.14	0.27	0.23	0.14
9	0.58	0.64	0.74	0.46	9	0.14	0.23	0.3	0.16
10	0.45	0.52	0.64	0.52	10	0.34	0.44	0.42	0.28
11	0.86	0.85	0.93	0.78	11	0.51	0.43	0.54	0.36
12	0.58	0.65	0.68	0.63	12	0.33	0.39	0.43	0.3
13	0.66	0.73	0.79	0.59	13	0.39	0.49	0.54	0.37
14	0.57	0.67	0.71	0.65	14	0.19	0.29	0.35	0.31
15	0.89	0.89	0.91	0.87	15	0.33	0.43	0.44	0.4
16	0.6	0.68	0.7	0.59	16	0.18	0.3	0.33	0.19
17	0.69	0.74	0.78	0.61	17	0.27	0.34	0.32	0.2
18	0.45	0.57	0.62	0.37	18	0.32	0.37	0.48	0.32
Across	0.6	0.66	0.66	0.63	Across	0.24	0.27	0.29	0.24
Average	0.63	0.69	0.73	0.62	Average	0.28	0.35	0.38	0.28
StDev	0.14	0.10	0.10	0.13	StDev	0.09	0.08	0.09	0.08

Table 9: Binary and multi-class classification performance of the DL models on all the labels. User 1 -18 show the performance within users while “across” highlights the performance across users

This study found that the hybrid model was the top-performing model overall with a binary classification accuracy of 73% and multi-class classification of 38%, across and within users. The hybrid model showed a dominant performance within users for binary classification, where it had the highest accuracy for 16 out of 18 users. Notably, the hybrid model had accuracies > 90% for users 11 and 15. EEGNet also had a good performance, beating the baseline accuracy across users for binary (58.79% vs 66%) and multi-class (25% vs 27%). EEGNet also had the joint highest binary classification across users and had the highest multi-class classification for 6 out of 18 users. DeepNet and ShallowNet performed a lot worse than the other models, where DeepNet had the highest accuracy for one user on the binary classification while ShallowNet had no case of highest accuracies.

Result 13: *Hybrid and EEGNet had a better performance than the baseline model overall.*

EEGNet and the Hybrid model were the only models to outperform the baseline model's performance with overall accuracies of 50% and 52% respectively (**Table 10**). The hybrid model's within user binary classification accuracy (74%) was the highest accuracy difference from the baseline and strongly contributed to the overall difference in binary classification (**Figure 21**). This was due to outperforming the baseline on both across user and within user predictions. However, ShallowNet and DeepNet failed to outperform the baseline model after both obtained an overall accuracy of 44%.

Evaluation Type	DeepNet	EEGNet	Hybrid	RF	ShallowNet
Across Users	0.42	0.47	0.48	0.41	0.43
binary	0.60	0.66	0.66	0.59	0.63
multi-class	0.24	0.27	0.29	0.24	0.24
Within Users	0.46	0.53	0.56	0.49	0.45
binary	0.63	0.69	0.74	0.66	0.62
multi-class	0.28	0.36	0.38	0.32	0.28
Average	0.44	0.50	0.52	0.45	0.44
StDev	0.03	0.04	0.06	0.05	0.01

Table 10: DL models vs baseline model (RF) classification performance on all labels.

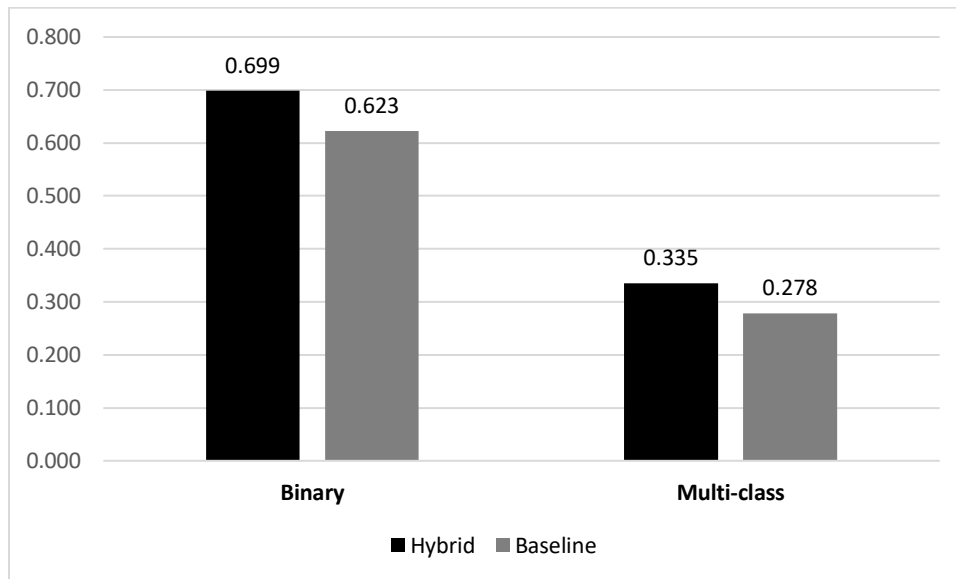


Figure 21: Hybrid model vs baseline model (RF) classification performance on all labels.

Results 14: Confusion matrices show that an attention level of 1 was the hardest to predict across users

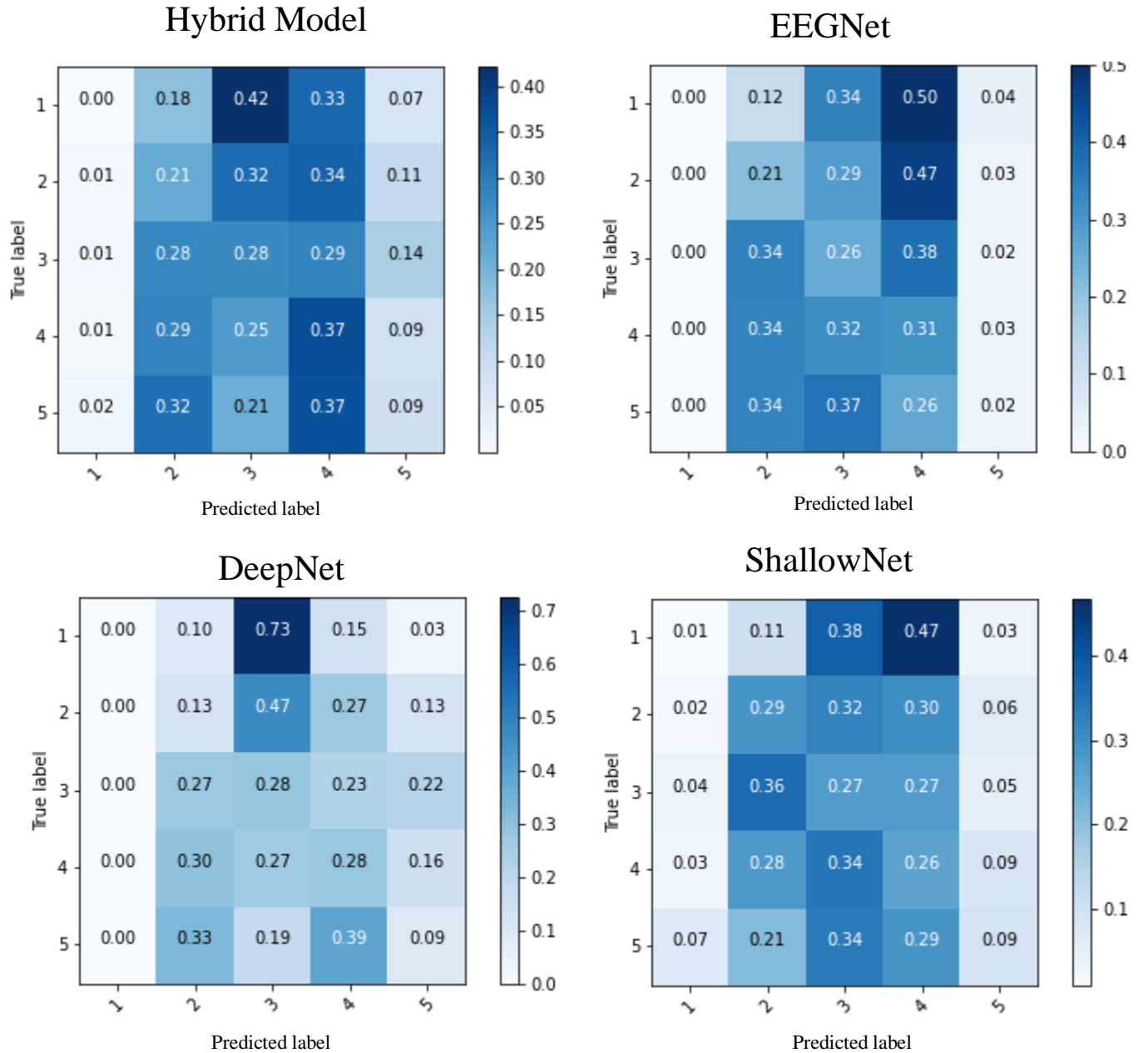


Figure 22: Confusion matrices of all the DL models across users for the attention label. Each number represents a score for attention. Accuracies were evaluated on the multi-class dataset.

From the confusion matrices, it is evident that barely any of the models were able to correctly predict the score of 1, where the highest accuracy was 0.01 (ShallowNet). This may be due to the high level of confusion with a score of 3, where the DeepNet misclassified it as a score of 3, 73% of the time. Attention score of 5 saw similar results, where the highest accuracy on that score was 0.09. Conversely, an attention score 4 saw the highest accuracy of any class, where the Hybrid model had an accuracy of 37% on that

class. Scores of 2 and 3 with slightly worse with accuracies on those classes ranging from 0.13-0.29 and 0.26 – 0.28 respectively.

5.4.6. Regularisation

Result 15: Early stopping improved accuracy

Early stopping reduced overfitting, as EEGNet and the Hybrid model both saw increases in accuracy (Figure 23A). Interestingly, early stopping reduced the number of epochs from 100 to 2 (Figure 23B).

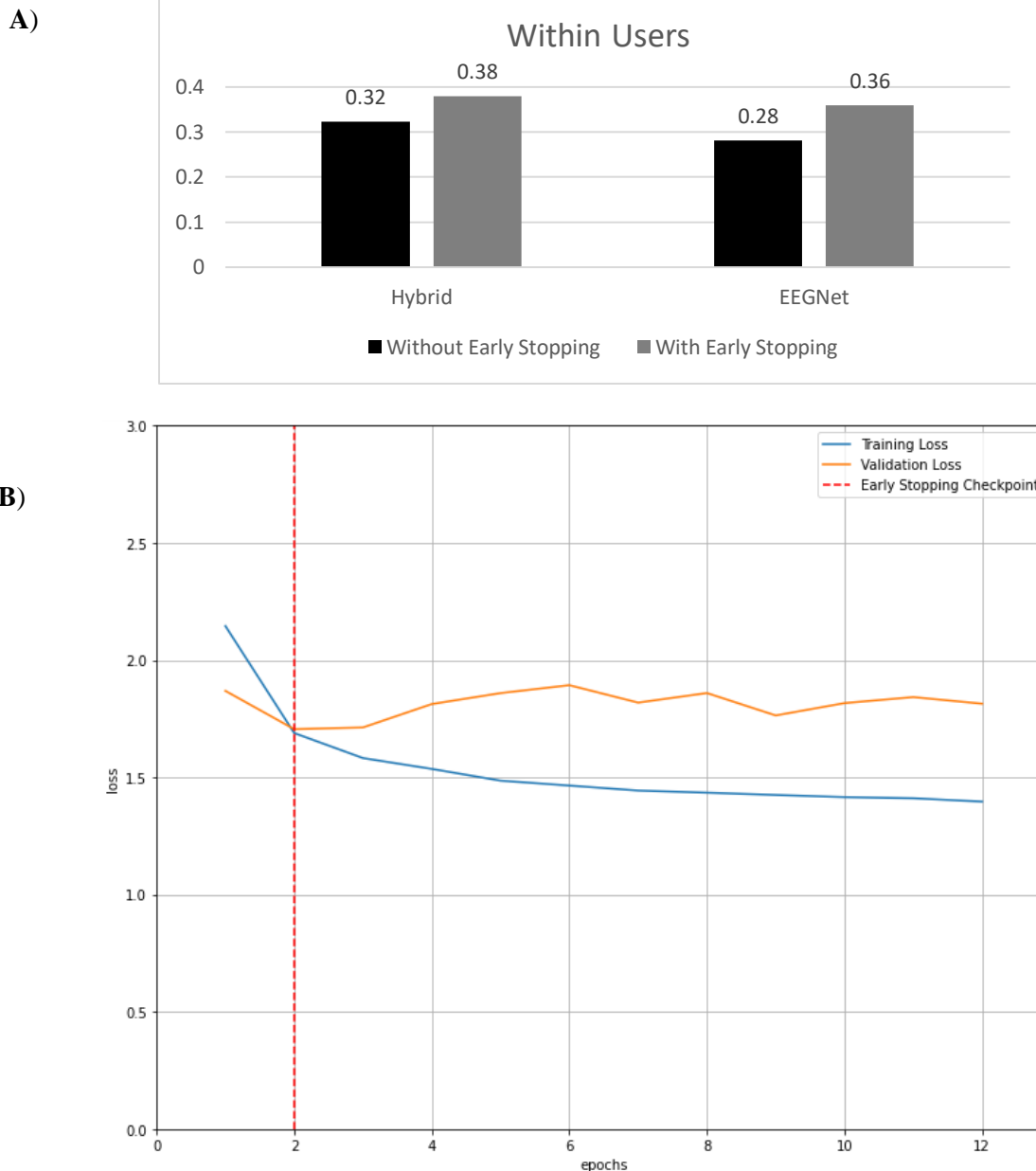


Figure 23: A) Bar graphs comparing Hybrid and EEGNet models’s multi-class performance with and without early stopping after predicting within users on all labels. Without early stopping the models were left to train for 100 epochs B) Plot showing the training and validation loss curves with early stopping for EEGNet

Result 16: *10% dropout of neurons yielded the highest accuracy*

This study ran an experiment to observe how dropout affected accuracy. The Hybrid model was used for the experiment and performance was evaluated across users on the multi-class attention dataset. From **Figure 24**, we can see that dropout had some interesting effects, as the lowest value (0.1) yielded the highest accuracy while the highest (0.9) produced the 2nd highest accuracy.

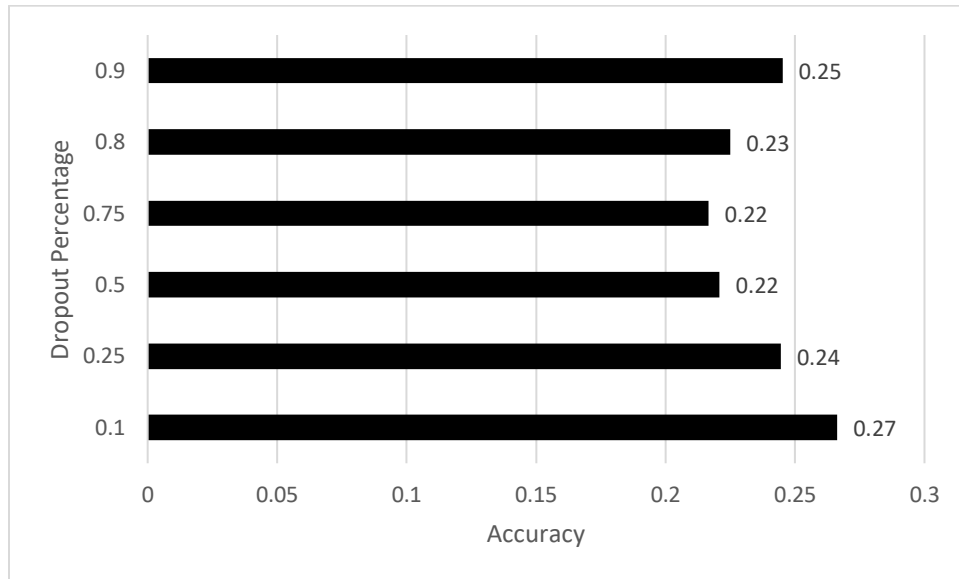


Figure 24: The effect of dropout on the Hybrid model. The performance was evaluated across users on the multi-class attention dataset

5.4.7 Training Strategy

Result 17: *Learning rate test helped to find the optimal learning rate*

This study ran a learning rate test to observe how changing the learning rate impacted performance. Indeed, there was a sharp increase in loss as the learning rate approached a value of 1. From observing the plot, this study tested learning rates of 0.1, 0.01 and 0.001 and found that a learning rate of 0.001 produced the best accuracy on the Hybrid model.

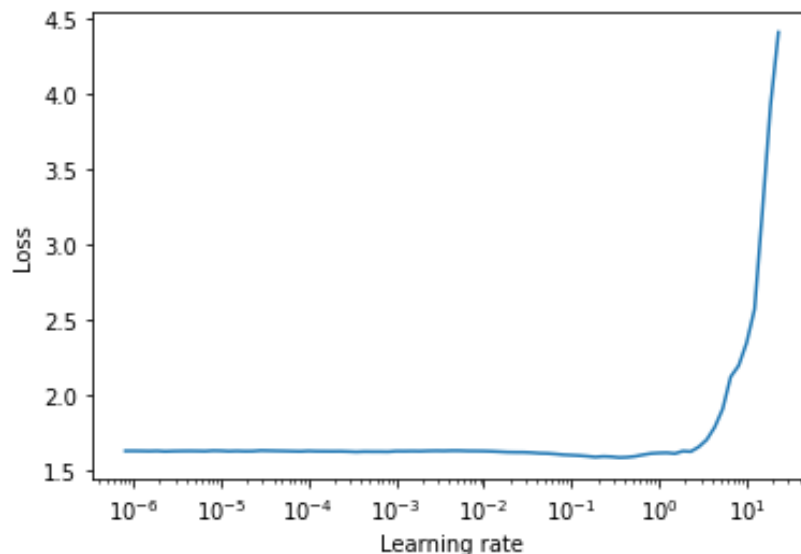


Figure 25: Plot showing the effect of learning rate on the Hybrid model.

5.5 Discussion and Critical Appraisal

5.5.1 Visualisation

The class frequency distribution plots (**Figure 15**), showed that scores of 1, 5 and 1 were the least frequent scores for attention, effort and interest, respectively. This significantly impacted the decoding accuracies of these classes as seen on the confusion matrices of the DL models in (**Figure 22**). An explanation for the low frequencies for scores 1 and 5, could be that participants were more inclined to give values centred around the mean than the extreme ones.

The correlation heatmap was insightful as it showed that EEG channels within users had a higher correlation with the targets labels than across users. This could be due to inter-subject variability which is a big challenge that faces a lot of EEG accuracy [4]. The decoding accuracies reflected this issue, as accuracies within users were consistently higher than across users. Moreover, there was a high correlation between the effort and attention labels, which could relate to how similar cognitive processes are engaged for both cognitively tasking and attention requiring activities [94].

Lastly, visualising the attention scores of User 1's reading sessions was very informative, showing that the EEG amplitudes of attention scores of 3,4,5 were similar while 1 and 2 were much lower. This could explain why binary classification saw a 30% increase in decoding accuracies over multi-class classification as the decision boundary was more prominent.

5.5.2 The Effect of Data Pre-Processing

This study found that changes in window size had an effect in decoding accuracies but a larger effect on run time. The increase run time was simply due to there being more samples to learn from. While the decoding accuracy increase could be a result of a larger window size like 120 samples, captured more information relating to its label. Indeed, future research will investigate larger window sizes to see if there is an upper limit.

Furthermore, bandpass filtering had mixed results. DeepNet and ShallowNet both saw increases in accuracy after the filters were applied while EEGNet and the Hybrid model saw no improvements. ShallowNet and DeepNet's increase could be due to having similar lower-level CNN architecture design choices unrelated to the depth of each network. The hybrid model is part composed of EEGNet hence why they both showed similar effects after the bandpass filters were applied.

5.5.3 Data Augmentation

Data augmentation only improved the decoding accuracies within users (0.38 to 0.399). The decrease in accuracy across uses may be caused by the inherent inter-subject variability and augmenting the data could have exacerbated this trait. Conversely, the increase in accuracy within users may be due to the model being able to generalise better on users with small sample sizes, as augmenting the data by a multiple of 20 saw the best performance.

5.5.4 Deep Learning Model Performance

Combining the spatial and temporal feature learning of CNNs and RNNs respectively proved beneficial, as the hybrid model was the top-performing model. The RNN certainly had an additive effect as the Hybrid model consistently outperformed the other CNN architectures (EEGNet, ShallowNet and DeepNet). These results were similar to other studies [13], [87], where hybrid CNN and RNN models outperformed CNNs.

Importantly, the hybrid model and EEGNet both outperformed the baseline model, suggesting that hierarchical feature learning helped to classify data that was minimally processed. However, ShallowNet and DeepNet performed poorly, and this may be due to differences in architecture design choices which may not have suited the data.

Lastly, the confusion matrices showed that attention scores of 1 and 5 were the most difficult to classify, which relates back to the issue of class imbalance.

5.5.5 Regularisation and Training Strategy

Early stopping was necessary to help reduce overfitting, which in turn led to an uplift in accuracy (**Figure 23**). As a result, the overall training for each model was reduced, allowing the study to efficiently carry out more experiments. Moreover, this study hypothesised that there would be a positive correlation with dropout percentage and accuracy; however, that was not the case. A low percentage of 10% yielded the best results, thus suggesting that further regularisation may not have been necessary with early stopping implemented. Lastly, the learning rate range test on the Hybrid model provided valuable information about the optimal learning rate.

Chapter 6: Conclusions

6.1 Introduction

This chapter summarises and reflects upon the project by first reviewing the objectives defined in **Section 1.4** and then discussing the future work for the project. Finally, this chapter concludes with an analysis of the limitations of the current project before ending with general reflections about what was accomplished and learned from this project.

6.2 Achievements

The primary objective of this project was to design a deep learning architecture for classifying EEG into scores of attention, interest and effort. To achieve this objective, this project designed a CNN architecture inspired by the Tensorflow implementation of EEGNet [11]. This study then went on to achieve the second primary objective by evaluating the classification results of EEGNet against existing EEG-specific CNN architectures such as ShallowNet and DeepNet using 5-fold cross-validation. Importantly, the secondary objective of the project was to design a variant deep learning architecture to improve the classification results. Indeed this project designed a hybrid deep learning model comprised of EEGNet and an RNN which outperformed the classification results of the other CNN architectures. Most notably, the hybrid model obtained an accuracy of 93% on user 11 with binary classification.

This study also went beyond the primary and secondary objectives to improve the classification results. This study converted the problem from multi-class to binary classification which led to a 30% increase in accuracy. Moreover, data augmentation provided a slight improvement in accuracy for within user analysis.

6.3 Future work

This study presents future research opportunities identified whilst completing this project, that could help improve the classification results.

6.3.1 EEG as an image

Given that most of the advances in deep learning have been in computer vision and there are a lot of image-pre-trained CNN models available, converting the EEG signals into a spectrogram for image classification could be a worthy research pursuit.

6.3.2 GANs-based data augmentation

This study did not consider GANs for data augmentation due to time constraints however, future research could be fruitful. Indeed Hartmann *et al.* [54] developed an EEG-specific GANS model called EEG-GAN, which could be used for augmentation the data.

6.3.3 Cyclical learning rates

This study used a static learning rate determined by the learning rate test, however research by Smith [95], has found that cyclical learning rates can yield better performance.

6.3.4 Class balancing strategies

Class balancing techniques, such as oversampling and undersampling, could help address the class imbalance problem identified in **Figure 15**. Future research could investigate oversampling the minority scores such as 1 and 5, with the use of SMOTE⁸[96] or random oversampling.

6.3.5 Variant DL architectures

The variant model designed comprised of a CNN (EEGNet) and an RNN. Other design architectures could be investigated like alternative compositions of RNNs and CNNs. More complex architectures could be investigated as seen in [13], where the authors build a model that consisted of a CNN, RNN, stacked autoencoder and XGBoost.

6.3.6 Variant bandpass filtering frequencies

This project filtered the EEG signals (0-40Hz) which led to mixed results across all the DL models. Future work could look into testing different range frequencies for the band-pass filter.

6.4 Limitations

The main limitation of this study was the impact of COVID-19. This forced this study to be done completely remotely, predominantly in isolation which made it challenging to collaborate and work productively. This could have negatively impacted the number of experiments that could have been done to improve the classification results further.

Moreover, the timestamps of the EEG signals were incorrectly recorded which made it difficult to calculate the actual frequency of the EEG signals as there was a discrepancy between the sampling frequency (500hz) and the number of samples per reading sessions. This may have adversely affected the bandpass filters, explaining why filtering the data gave mixed results.

6.5 Concluding Remarks

This study designed a novel hybrid deep learning architecture for classifying EEG signals into scores for attention, interest and effort. Besides achieving all the objectives laid out, this study also provided a proof of concept for EEG-based attention detection. This proof of concept can be built upon by future research to develop brain-computer interface (BCI) applications for detecting attention.

Finally, the learning outcomes for me from this project were highly rewarding, as this project required me to research extensively on EEG, attention and deep learning. The practical element of working with Python and PyTorch to build a deep learning project was also worthwhile, as this experience can be applied to many other domains.

⁸ Synthetic Minority Oversampling Technique

Chapter 7: Ethics

This study already has prior ethical approval. The data used was anonymised and the ethics for obtaining the original data has already been approved. Please refer to **Appendix D** for the ethical approval letter

References

- [1] M. A. Moreno Rocha, N. Miguel, W. McCaffery, J. Ye, Y. Lei, and W. Z. (Creator), “Instrumented Digital and Paper Reading (dataset),” 2019. [Online]. Available: [https://risweb.st-andrews.ac.uk/portal/en/datasets/instrumented-digital-and-paper-reading-dataset\(80f522b6-6d23-4751-9023-21a1e3d0eb5a\).html](https://risweb.st-andrews.ac.uk/portal/en/datasets/instrumented-digital-and-paper-reading-dataset(80f522b6-6d23-4751-9023-21a1e3d0eb5a).html).
- [2] N. Bigdely-Shamlo, T. Mullen, C. Kothe, K. M. Su, and K. A. Robbins, “The PREP pipeline: Standardized preprocessing for large-scale EEG analysis,” *Front. Neuroinform.*, vol. 9, no. JUNE, pp. 1–19, 2015, doi: 10.3389/fninf.2015.00016.
- [3] M. Jas, D. A. Engemann, Y. Bekhti, F. Raimondo, and A. Gramfort, “Autoreject: Automated artifact rejection for MEG and EEG data,” *Neuroimage*, vol. 159, pp. 417–429, 2017.
- [4] Y. Roy, H. Banville, I. Albuquerque, A. Gramfort, T. H. Falk, and J. Faubert, “Deep learning-based electroencephalography analysis: A systematic review,” *J. Neural Eng.*, vol. 16, no. 5, 2019, doi: 10.1088/1741-2552/ab260c.
- [5] S. R. Cole and B. Voytek, “Cycle-by-cycle analysis of neural oscillations. bioRxiv.” 2018.
- [6] A. Gramfort, D. Strohmeier, J. Haueisen, M. S. Hämäläinen, and M. Kowalski, “Time-frequency mixed-norm estimates: Sparse M/EEG imaging with non-stationary source activations,” *Neuroimage*, vol. 70, pp. 410–422, 2013.
- [7] M. Clerc, L. Bougrain, and F. Lotte, “Brain-computer interfaces,” 2016.
- [8] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [9] Y. LeCun, “The MNIST database of handwritten digits,” <http://yann.lecun.com/exdb/mnist/>, 1998.
- [10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*, 2009, pp. 248–255.
- [11] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, “EEGNet: A Compact Convolutional Network for EEG-based Brain-Computer Interfaces,” Nov. 2016, doi: 10.1088/1741-2552/aace8c.
- [12] R. T. Schirrmeister *et al.*, “Deep learning with convolutional neural networks for EEG decoding and visualization,” Mar. 2017, doi: 10.1002/hbm.23730.
- [13] X. Zhang, L. Yao, Q. Z. Sheng, S. S. Kanhere, T. Gu, and D. Zhang, “Converting Your Thoughts to Texts: Enabling Brain Typing via Deep Feature Learning of EEG Signals,” *2018 IEEE Int. Conf. Pervasive Comput. Commun. PerCom 2018*, pp. 1–10, 2018, doi:

10.1109/PERCOM.2018.8444575.

- [14] I. Goodfellow, "NIPS 2016 tutorial: Generative adversarial networks," *arXiv Prepr. arXiv1701.00160*, 2016.
- [15] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Advances in neural information processing systems*, 2007, pp. 137–144.
- [16] P. F. Diez, A. G. Correa, L. Orosco, E. Laciari, and V. Mut, "Attention-level transitory response: a novel hybrid BCI approach," *J. Neural Eng.*, vol. 12, no. 5, p. 56007, 2015.
- [17] E. B. Goldstein, *Cognitive Psychology. CONNECTING MIND, RESEARCH, AND EVERYDAY EXPERIENCE*. 2011.
- [18] R. Parasuraman and P. G. Nestor, "Attention and driving skills in aging and Alzheimer's disease," *Hum. Factors*, vol. 33, no. 5, pp. 539–557, 1991.
- [19] K. Murkett, W. Smart, and K. Nugent, "Attention-deficit/hyperactivity disorder in postsecondary students," *Neuropsychiatr. Dis. Treat.*, p. 1781, 2014, doi: 10.2147/ndt.s64136.
- [20] Y. Li, X. Li, L. Liu, Q. Liu, Y. Qi, and M. Ratcliffe, *A Real-time EEG-based BCI System for Attention Recognition in Ubiquitous Environment*. ACM, 2011.
- [21] S. Sanei and J. A. Chambers, *EEG signal processing*. John Wiley & Sons, 2013.
- [22] S. Noachtar, C. Binnie, J. Ebersole, F. Mauguier, A. Sakamoto, and B. Westmoreland, "A glossary of terms most commonly used by clinical electroencephalographers and proposal for the report form for the EEG findings. The International Federation of Clinical Neurophysiology.," *Electroencephalogr. Clin. Neurophysiol. Suppl.*, vol. 52, p. 21, 1999.
- [23] C. Babiloni *et al.*, "International Federation of Clinical Neurophysiology (IFCN)--EEG research workgroup: Recommendations on frequency and topographic analysis of resting state EEG rhythms. Part 1: Applications in clinical research studies," *Clin. Neurophysiol.*, vol. 131, no. 1, pp. 285–307, 2020.
- [24] M. Alirezaei and S. Sardouie, *Detection of Human Attention Using EEG Signals*. 2017.
- [25] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning."
- [26] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, 1986, doi: 10.1038/323533a0.
- [27] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," 2011.
- [28] A. Géron, "Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems SECOND EDITION," 2019.
- [29] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," 2012, doi: 10.1109/CVPR.2012.6248110.
- [30] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553. Nature Publishing Group, pp. 436–444, May 27, 2015, doi: 10.1038/nature14539.
- [31] A. Mittal, "Understanding RNN and LSTM," *Toward. Data Sci.*, 2019, [Online]. Available: <https://towardsdatascience.com/understanding-rnn-and-lstm-f7cdf6dfc14e>.
- [32] A. Amidi and S. Amidi, "Cheatsheet recurrent neural networks," *Stanford*, [Online]. Available: <https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks#architecture>.

- [33] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, 1997, doi: 10.1162/neco.1997.9.8.1735.
- [34] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *J. Mach. Learn. Res.*, 2003, doi: 10.1162/153244303768966139.
- [35] J. Schmidhuber, "Deep Learning in neural networks: An overview," *Neural Networks*. 2015, doi: 10.1016/j.neunet.2014.09.003.
- [36] R. Srinivasan, S. Thorpe, S. Deng, T. Lappas, and M. D'Zmura, "Decoding Attentional Orientation from EEG Spectra," in *Human-Computer Interaction. New Trends*, 2009, pp. 176–183.
- [37] D. W. Beebe, D. Rose, and R. Amin, "Attention, learning, and arousal of experimentally sleep-restricted adolescents in a simulated classroom," *J. Adolesc. Heal.*, vol. 47, no. 5, pp. 523–525, 2010.
- [38] W. J. Ray and H. W. Cole, "EEG alpha activity reflects attentional demands, and beta activity reflects emotional and cognitive processes," *Science (80-.)*, vol. 228, no. 4700, pp. 750–752, 1985.
- [39] T.-P. Jung, S. Makeig, M. Stensmo, and T. J. Sejnowski, "Estimating alertness from the EEG power spectrum," *IEEE Trans. Biomed. Eng.*, vol. 44, no. 1, pp. 60–69, 1997.
- [40] K. Yaomanee, S. Pan-ngum, and P. I. N. Ayuthaya, "Brain signal detection methodology for attention training using minimal EEG channels," in *2012 Tenth International Conference on ICT and Knowledge Engineering*, 2012, pp. 84–89.
- [41] N. H. Liu, C. Y. Chiang, and H. C. Chu, "Recognizing the degree of human attention using EEG signals from mobile sensors," *Sensors (Switzerland)*, vol. 13, no. 8, pp. 10273–10286, 2013, doi: 10.3390/s130810273.
- [42] B. Hamadicharef *et al.*, "Learning EEG-based spectral-spatial patterns for attention level measurement," in *Proceedings - IEEE International Symposium on Circuits and Systems*, 2009, pp. 1465–1468, doi: 10.1109/ISCAS.2009.5118043.
- [43] S. Aliakbaryhosseinabadi, E. Nlandu Kamavuako, N. Jiang, D. Farina, and N. Mrachacz-Kersting, "Classification of EEG signals to identify variations in attention during motor task 1 execution 2."
- [44] B. Hu, X. Li, S. Sun, and M. Ratcliffe, "Attention Recognition in EEG-Based Affective Learning Research Using CFS+KNN Algorithm," *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 15, no. 1, pp. 38–45, 2018, doi: 10.1109/TCBB.2016.2616395.
- [45] L. Deckers, N. Das, A. H. Ansari, A. Bertrand, and T. Francart, "EEG-based detection of the attended speaker and the locus of auditory attention with convolutional neural networks," *bioRxiv*, p. 475673, 2018.
- [46] S. Borhani, R. Abiri, J. Muhammad, Y. Jiang, and X. Zhao, "EEG-based Visual Attentional State Decoding Using Convolutional Neural Network," 2016.
- [47] R. Rao and R. Derakhshani, "A comparison of EEG preprocessing methods using time delay neural networks," *2nd Int. IEEE EMBS Conf. Neural Eng.*, vol. 2005, pp. 262–264, 2005, doi: 10.1109/CNE.2005.1419607.
- [48] R. Abiri, X. Zhao, and Y. Jiang, "A real time EEG-based neurofeedback platform for attention training," 2016.

- [49] A. Mogron, J. Jovicich, L. Bruzzone, and M. Buiatti, "ADJUST: An automatic EEG artifact detector based on the joint use of spatial and temporal features," *Psychophysiology*, vol. 48, no. 2, pp. 229–240, 2011.
- [50] H. Nolan, R. Whelan, and R. B. Reilly, "FASTER: fully automated statistical thresholding for EEG artifact rejection," *J. Neurosci. Methods*, vol. 192, no. 1, pp. 152–162, 2010.
- [51] V. Lawhern, W. D. Hairston, and K. Robbins, "DETECT: A MATLAB toolbox for event detection and identification in time series, with applications to artifact detection in EEG signals," *PLoS One*, vol. 8, no. 4, 2013.
- [52] X. Li, P. Zhang, D. Song, G. Yu, Y. Hou, and B. Hu, "EEG based emotion identification using unsupervised deep feature learning," 2015.
- [53] Y. Ren and Y. Wu, "Convolutional deep belief networks for feature extraction of EEG signal," *Proc. Int. Jt. Conf. Neural Networks*, pp. 2850–2853, 2014, doi: 10.1109/IJCNN.2014.6889383.
- [54] K. G. Hartmann, R. T. Schirrmester, and T. Ball, "EEG-GAN: Generative adversarial networks for electroencephalographic (EEG) brain signals," *arXiv Prepr. arXiv1806.01875*, 2018.
- [55] I. Sturm, S. Lapuschkin, W. Samek, and K.-R. Müller, "Interpretable deep neural networks for single-trial EEG classification," *J. Neurosci. Methods*, vol. 274, pp. 141–145, 2016.
- [56] J. Teo, C. L. Hou, and J. Mountstephens, "Preference classification using Electroencephalography (EEG) and deep learning," *J. Telecommun. Electron. Comput. Eng.*, 2018.
- [57] G. Ruffini *et al.*, "Deep learning with EEG spectrograms in rapid eye movement behavior disorder," *Front. Neurol.*, vol. 10, no. JUL, pp. 1–9, 2019, doi: 10.3389/fneur.2019.00806.
- [58] U. R. Acharya, S. V. Sree, A. P. C. Alvin, and J. S. Suri, "Use of principal component analysis for automatic classification of epileptic EEG activities in wavelet framework," *Expert Syst. Appl.*, vol. 39, no. 10, pp. 9072–9078, 2012.
- [59] D. A. V. Dyk and X. L. Meng, "The art of data augmentation," *J. Comput. Graph. Stat.*, 2001, doi: 10.1198/10618600152418584.
- [60] J. L. Perez-Benitez, J. A. Perez-Benitez, and J. H. Espina-Hernandez, "Development of a brain computer interface interface using multi-frequency visual stimulation and deep neural networks," 2018, doi: 10.1109/CONIELECOMP.2018.8327170.
- [61] S. Yang, S. Lopez, M. Golmohammadi, I. Obeid, and J. Picone, "Semi-automated annotation of signal events in clinical EEG data," 2017, doi: 10.1109/SPMB.2016.7846855.
- [62] F. Wang, S. H. Zhong, J. Peng, J. Jiang, and Y. Liu, "Data augmentation for eeg-based emotion recognition with deep convolutional neural networks," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, vol. 10705 LNCS, pp. 82–93, doi: 10.1007/978-3-319-73600-6_8.
- [63] M. Soleymani, J. Lichtenauer, T. Pun, and M. Pantic, "A multimodal database for affect recognition and implicit tagging," *IEEE Trans. Affect. Comput.*, 2012, doi: 10.1109/T-AFFC.2011.25.
- [64] W. L. Zheng and B. L. Lu, "Investigating Critical Frequency Bands and Channels for EEG-Based Emotion Recognition with Deep Neural Networks," *IEEE Trans. Auton. Ment. Dev.*, 2015, doi: 10.1109/TAMD.2015.2431497.
- [65] Y. Lecun, L. Bottou, Y. Bengio, and P. Ha, "LeNet," *Proc. IEEE*, 1998, doi: 10.1109/5.726791.

- [66] S. Wu, S. Zhong, and Y. Liu, "ResNet," *Multimed. Tools Appl.*, 2017, doi: 10.1007/s11042-017-4440-4.
- [67] I. Ullah, M. Hussain, E. ul H. Qazi, and H. Aboalsamh, "An automated system for epilepsy detection using EEG brain signals based on deep learning approach," *Expert Syst. Appl.*, 2018, doi: 10.1016/j.eswa.2018.04.021.
- [68] A. O'Shea, G. Lightbody, G. Boylan, and A. Temko, "Neonatal seizure detection using convolutional neural networks," 2017, doi: 10.1109/MLSP.2017.8168193.
- [69] N. S. Kwak, K. R. Müller, and S. W. Lee, "A convolutional neural network for steady state visual evoked potential classification under ambulatory environment," *PLoS One*, 2017, doi: 10.1371/journal.pone.0172578.
- [70] J. T. C. Schwabedal, J. C. Snyder, A. Cakmak, S. Nemati, and G. D. Clifford, "Addressing Class Imbalance in Classification Problems of Noisy Signals by using Fourier Transform Surrogates," *arXiv Prepr. arXiv1806.08675*, 2018.
- [71] A. Vilamala, K. H. Madsen, and L. K. Hansen, "Deep convolutional neural networks for interpretable analysis of EEG sleep stage scoring," 2017, doi: 10.1109/MLSP.2017.8168133.
- [72] S. Chambon, M. N. Galtier, P. J. Arnal, G. Wainrib, and A. Gramfort, "A Deep Learning Architecture for Temporal Sleep Stage Classification Using Multivariate and Multimodal Time Series," *IEEE Trans. Neural Syst. Rehabil. Eng.*, 2018, doi: 10.1109/TNSRE.2018.2813138.
- [73] K. W. Ha and J. W. Jeong, "Motor imagery EEG classification using capsule networks," *Sensors (Switzerland)*, vol. 19, no. 13, 2019, doi: 10.3390/s19132854.
- [74] J. Behncke, R. T. Schirrmeister, W. Burgard, and T. Ball, "The signature of robot action success in EEG signals of a human observer: Decoding and visualization using deep convolutional neural networks," 2018, doi: 10.1109/IWW-BCI.2018.8311531.
- [75] R. Manor and A. B. Geva, "Convolutional neural network for multi-category rapid serial visual presentation BCI," *Front. Comput. Neurosci.*, 2015, doi: 10.3389/fncom.2015.00146.
- [76] A. Craik, Y. He, and J. L. Contreras-Vidal, "Deep learning for electroencephalogram (EEG) classification tasks: a review," *J. Neural Eng.*, vol. 16, no. 3, p. 031001, Jun. 2019, doi: 10.1088/1741-2552/ab0ab5.
- [77] S. Kaur, "Noise Types and Various Removal Techniques," *International J. Adv. Res. Electron. cs Commun. Eng.*, 2015.
- [78] R. O. Duda, P. E. Hart, and D. G. Stork, "Pattern Classification (2nd ed .)," *Comput. Complex.*, 1998.
- [79] N. Mohanty, A. L.-S. John, R. Manmatha, and T. M. Rath, "Shape-Based Image Classification and Retrieval," in *Handbook of Statistics*, vol. 31, Elsevier, 2013, pp. 249–267.
- [80] H. Zhang, "The optimality of Naive Bayes," 2004.
- [81] Sklearn, "Naive Bayes," 2020, [Online]. Available: https://scikit-learn.org/stable/modules/naive_bayes.html.
- [82] G. Ke *et al.*, "LightGBM: A highly efficient gradient boosting decision tree," *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, no. Nips, pp. 3147–3155, 2017.
- [83] C. Chen, A. Liaw, and L. Breiman, "Using Random Forest to Learn Imbalanced Data," *Discovery*,

no. 1999, pp. 1–12, 2004.

- [84] T. K. Ho, “Random decision forests,” in *Proceedings of 3rd international conference on document analysis and recognition*, 1995, vol. 1, pp. 278–282.
- [85] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, 2017, doi: 10.1145/3065386.
- [86] L. H. Shi Xinjie, Wang Tianqi, Wang Lan, *Hybrid Convolutional Recurrent Neural Networks Outperform CNN and RNN in Task-state EEG Detection for Parkinson’s Disease*. 2019.
- [87] D. A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and accurate deep network learning by exponential linear units (ELUs),” 2016.
- [88] Hobs, “How to choose the number of hidden layers and nodes in a feedforward neural network?,” 2019, [Online]. Available: <https://stats.stackexchange.com/questions/181/how-to-choose-the-number-of-hidden-layers-and-nodes-in-a-feedforward-neural-net/1097#1097>.
- [89] H. B. Demuth, M. H. Beale, O. De Jess, and M. T. Hagan, *Neural network design*. Martin Hagan, 2014.
- [90] R. T. Schirrmeister *et al.*, “Deep learning with convolutional neural networks for EEG decoding and visualization,” *Hum. Brain Mapp.*, vol. 38, no. 11, pp. 5391–5420, 2017, doi: 10.1002/hbm.23730.
- [91] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” 2015.
- [92] Z. Mohamed, M. El Halaby, T. Said, D. Shawky, and A. Badawi, “Characterizing focused attention and working memory using EEG,” *Sensors (Switzerland)*, 2018, doi: 10.3390/s18113743.
- [93] L. N. Smith, “Cyclical learning rates for training neural networks,” *Proc. - 2017 IEEE Winter Conf. Appl. Comput. Vision, WACV 2017*, no. April, pp. 464–472, 2017, doi: 10.1109/WACV.2017.58.
- [94] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: synthetic minority over-sampling technique,” *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.
- [95] A. Rayome, “Python is the real language for data science,” 2018. <https://www.techrepublic.com/article/why-python-is-the-real-language-of-data-science-not-r/>.
- [96] Rayome, “Report: 59% of employed data scientists learned skills on their own or via a MOOC,” 2017, [Online]. Available: <https://www.techrepublic.com/article/report-59-of-employed-data-scientists-learned-skills-on-their-own-or-via-a-mooc/>.

Appendices

A. User Manual

1. Clone this repository - <https://github.com/Khalizo/Deep-Learning-Detection-Of-EEG-Based-Attention.git>
2. The directory contains requirements.txt file containing the required libraries:
`cd Deep-Learning-Detection-Of-EEG-Based-Attention`
`pip install -r requirements.txt`
3. Download the dataset via the one drive link. - https://universityofstandrews907-my.sharepoint.com/:u:/g/personal/ybk1_st-andrews_ac_uk/EYZMKRYhN_RPpekFwwjuTCsBO3dYQQbHbBcF6Newp36_PA?e=3sSmB0
4. Unzip and move the dataset to src
5. Run either baseline, EEGNet_Hybrid or ShallowDeep notebooks to observe the performance

Here is how to run each of the models:

NB: ShallowNet and DeepNet take long to run so if you want to see quick results, best to run EEGNet_Hybrid or baseline

ShallowNet and DeepNet:

- Simply run the notebook "Shallow Deep", and it will run 5 fold cross-validation for both models across users with binary classification
- To change the settings, right at the bottom you can change to "within users" by changing "cross user" to "per user" or change from binary to "multi" for multi-class classification

EEGNET_Hybrid

- Run EEGNet_Hybrid notebook The Default parameters are set to: Hybrid model running binary classification cross users

Baselines

- Run the baseline notebook, and it will run five-fold CV on all the models

Results

- Results are dumped in the results folder

B. Programming Language and Deep Learning Framework

This section was used to justify the programming language used and the deep learning framework.

B.1 Programming Language

This project used Python as the main programming language for the following reasons:

- It is the most popular language for deep learning according to a GitHub [97] and a recent survey conducted by Kaggle [98]
- It has useful machine learning libraries such as PyTorch, TensorFlow, Sklearn and Theano
- Wide community and support
- For EEG analysis, Python also has libraries such as PyEEG and Braindecode
- Moderate learning curve
- Easy to create prototypes
- Good visualisation options- matplotlib, seaborn, gplot
- Simplicity

A.2 Deep Learning Framework

This project used PyTorch as the deep learning framework. A study by Gradient showed that PyTorch has is on a steeper trajectory in terms of adoption in the research community (based on the number of papers implemented at major conferences (CVPR, ICRL, ICML, NIPS, ACL, ICCV etc.). As you can see from the data, in 2018 PyTorch was clearly a minority, compared with 2019 it's overwhelmingly favored by researchers at major conferences (Figure). Moreover, PyTorch is very pythonic and easy to debug (table) and the Braindecode library for EEG analysis runs on Pytorch.

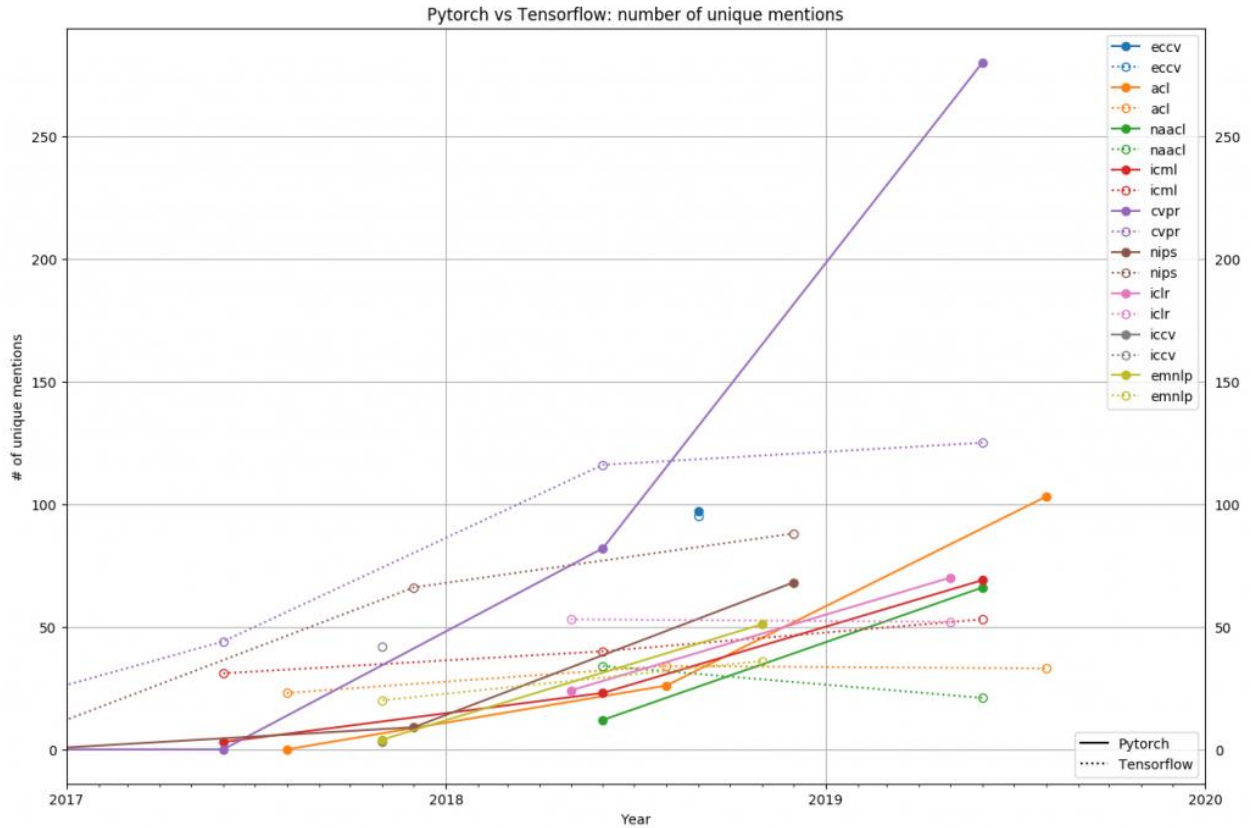


Figure 26: Graph Showing PyTorch's research adoption over time

Feature	PyTorch	TensorFlow
Release year	2016	2015
Programming API	Very pythonic and easy to debug	Cryptic to start with and can be hard to debug
Computation graph	Static computation graph where one defines the sequence of computation that one wants to do	Dynamic computation graph approach, where computations are done line by line
Distributed Computing	Can run on single/multiple distributed CPUs or GPUs	Can run on single/multiple distributed CPUs or GPUs
Deployment/Production	Deploys models with TorchServe but yet to mature	Excellent deployment ability due to its Static computation graph approach and also packages for quick deployment

Table 11: Comparison between PyTorch and TensorFlow

C. Clean Data Set Information

User	Number of rows of EEG Data
1	85991
2	155966
3	145361
4	171763
5	186316
6	222319
7	268922
8	35673
9	82768
10	11120
11	107552
12	177374
13	48850
14	142794
15	138960
16	236532
17	109652
18	261336
Total	2589249

Table 12: Table showing the amount of EEG data per user

D. Ethical Application Approval Letter

DocuSign Envelope ID: E6BB71C5-24FF-4F12-9D11-AAEA53F3E0D7

UNIVERSITY OF ST ANDREWS
TEACHING AND RESEARCH ETHICS COMMITTEE (UTREC)
SCHOOL OF COMPUTER SCIENCE
PRELIMINARY ETHICS SELF-ASSESSMENT FORM

This Preliminary Ethics Self-Assessment Form is to be conducted by the researcher, and completed in conjunction with the Guidelines for Ethical Research Practice. All staff and students of the School of Computer Science must complete it prior to commencing research.

This Form will act as a formal record of your ethical considerations.

Tick one box

- ☐ **Staff Project**
☒ **Postgraduate Project**
☐ **Undergraduate Project**

Title of project

Deep Learning Detection of EEG Based Attention

Name of researcher(s)

Babs Khalidson

Name of supervisor (for student research)

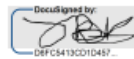
Dr Juan Ye

OVERALL ASSESSMENT (to be signed after questions, overleaf, have been completed)

Self audit has been conducted **YES** ☒ **NO** ☐

There are no ethical issues raised by this project

Signature Student or Researcher


DocuSigned by:
DEF6M413C0D0457...

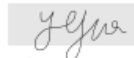
Print Name

Babs Khalidson

Date

29/05/2020

Signature Lead Researcher or Supervisor



Print Name

Dr Juan Ye

Date

5/30/2020

This form must be date stamped and held in the files of the Lead Researcher or Supervisor. If fieldwork is required, a copy must also be lodged with appropriate Risk Assessment forms. The School Ethics Committee will be responsible for monitoring assessments.

Computer Science Preliminary Ethics Self-Assessment Form

Research with human subjects

Does your research involve human subjects or have potential adverse consequences for human welfare and wellbeing?

YES ☐ NO ☒

If YES, full ethics review required

For example:

Will you be surveying, observing or interviewing human subjects?

Will you be analysing secondary data that could significantly affect human subjects?

Does your research have the potential to have a significant negative effect on people in the study area?

Potential physical or psychological harm, discomfort or stress

Are there any foreseeable risks to the researcher, or to any participants in this research?

YES ☐ NO ☒

If YES, full ethics review required

For example:

Is there any potential that there could be physical harm for anyone involved in the research?

Is there any potential for psychological harm, discomfort or stress for anyone involved in the research?

Conflicts of interest

Do any conflicts of interest arise?

YES ☐ NO ☒

If YES, full ethics review required

For example:

Might research objectivity be compromised by sponsorship?

Might any issues of intellectual property or roles in research be raised?

Funding

Is your research funded externally?

YES ☐ NO ☒

If YES, does the funder appear on the 'currently automatically approved' list on the UTREC website?

YES ☐ NO ☐

If NO, you will need to submit a Funding Approval Application as per instructions on the UTREC website.

Research with animals

Does your research involve the use of living animals?

YES ☐ NO ☒

If YES, your proposal must be referred to the University's Animal Welfare and Ethics Committee (AWEC)

University Teaching and Research Ethics Committee (UTREC) pages

<http://www.st-andrews.ac.uk/utrec/>